PERSONAL COMPUTER MAGAZINE

BITS & BYTES

1-9826

Issue No. 7, April 1983: $1.00

Micro networks for schools

NZ-made disk drives for System 80/TRS 80

Winning Sinclair program

New Sord column
plus our usual columns

More books from our book club

Two P.C.'s reviewed

IBM PC
NEC PC8000

IN JAPAN THE NEC PC-8000 OUTSELLS ITS NEAREST COMPETITOR BY 2 TO 1

HERES WHY...

The PC-8000 Personal Computer for Professionals from NEC. Prices including a high resolution monitor start at $2140. The PC-8000 is a highly reliable personal computer ideal for applications in business and the professions. It features a powerful BASIC, many versatile applications packages, excellent screen graphics, and high ease of use. The PC-8000 can be used as a stand-alone computer or as a terminal attached to a host computer.

Either a PC-8023A Dot Matrix or a letter quality stand alone typewriter can be attached to the PC-8000.

THE FACTS

PC 8001B

PROCESSOR: μPD780C-1 (Z-80A compatible, 4 MHz).

MAIN MEMORY: 32K (32,768) bytes Dynamic RAM Memory. Readily extendable to 64 K or for special applications up to 192 K.

ROM MEMORY: 24 K (24,576) bytes minimum including Microsoft Basic and Machine Code Monitor.

KEYBOARD: 82 keys typewriter style with 20-key numeric pad, upper/lower case graphic and scientific characters, and control keys. — 5 programmable function keys × 2 functions each.

CLOCK: Time and date can be set and read from programs. Keeps track of year, month, day and the exact time.

ATTRIBUTES: User Programmable Screen Size 36, 40, 72 or 80 characters by 20 or 25 lines. 248 characters and graphic symbols 160 × 100 dots graphic mode 8 colours (black, blue, red, magenta, green, cyan, yellow or white) or intensities, and screen attributes (blinking, reverse, hidden, etc.) Built in speaker.

INTERFACES: FSK system (1200, 2400 Hz), 600 baud. Standard 8-bit parallel printer interface. Serial interface with optional cable. Color (RGB) Video and Monochrome (Composite) interfaces. Z80 Processor bus interface.

TERMINAL MODE: Allows the PC to operate as an ASCII terminal. The terminal mode can be entered from a program or by the operator. Software emulators available for most IBM and other mainframes.

PERIPHERALS

EXPANSION UNIT: Various Expansion Units available for: Floppy and/or Fixed Disk, Memory Expansion, Real Time Clocks with priority interrupts; Single bit input and output ports, Addition Printer RS232C, and IEEE 488 interfaces. Card slots available for user defined PCBs.

DOT MATRIX PRINTER: Tractor feed or friction feed. 100 characters/sec, bidirectional upper/lower case characters, numerals, symbols compressed, and double width character expansion; true descenders; 310-character symbol set (matches ROM character set). Proportional spacing.

MASS STORAGE: Dual 5.25 or 8 inch floppy disk drives giving 655 Kbytes or 2.5 Mbytes storage capacity. Corvus Winchester disks at 6, 11 or 20 Mbytes available. Multiplexors allow sharing of 5.25 inch or Corvus disk units between several machines.

DISPLAY: High Resolution 12" Green Phosphor, High Resolution (2000 pixel) Color Display.

OTHER: Light Pens, Flat Bed Plotters and other peripherals available.

PROGRAMMING ENVIRONMENT

DOS: NEC DOS or (optional) CP/M®, UCSD p® system, Racet® DOS etc.

LANGUAGES: N-BASIC (Microsoft) and Assembler or (optional) Fortran, Cobol, PL/Z Pascal, Lisp, Basic Compilers.

AVAILABLE SOFTWARE: Numerous Packages available immediately including Benchmark® System Word Processor, Mailing List Manager. Various Accounting Packages including NZ requirements. Farming Packages. Super-Calc®, Report Maker®, T/Maker® etc electronic spread-sheets. Business Planning Packages. Database Systems. Telecommunication Protocols. Secretarial Packages.

NEC

Nippon Electric Co Ltd

Tokyo Japan

COMPLETE LIST OF DEALERS ON FACING PAGE.

SCOLLAY COMPUTERS

(A Division of Wm Scollay & Company Ltd)

4th Floor, DFC Bldg, Cnr Grey & Featherston Sts, Wellington, P.O. Box 2377.

FOR FURTHER DETAILS CONTACT YOUR NEAREST DEALER

Aztec Management Services
29-33 College Hill
PO Box 47-182
AUCKLAND
Phone: (09) 793-496

D E & J Goldfinch Ltd
Highbury House
65-67 Birkenhead Ave
PO Box 34-196
AUCKLAND 10
Phone: (09) 483-342

Software Architects Ltd
10 Heather Street, Parnell
PO Box 9652, Newmarket
AUCKLAND
Phone: (09) 732-427

A1 Computer Services Ltd
Kings Arcade, Victoria St
PO Box 1188
HAMILTON
Phone: (071) 393-000

Waikato Computer Centre
6 Princes Street
PO Box 1094
HAMILTON
Phone: (071) 393-416

Taranaki Micro Electronics Ltd
Centre Court Building
Devon Street East
NEW PLYMOUTH
Phone: (067) 84-067

Pattersons Office Systems
211 Queen Street East
PO Box 979
HASTINGS
Phone: (070) 85-161

David Brice Electronics Ltd
47 Kimbolton Road
FEILDING
Phone: (063) 37-141

Hands On Information Systems
35-37 Victoria Street
WELLINGTON
Phone: (04) 725-224

Computers for People
35 Taranaki Street
WELLINGTON
Phone: (04) 847-668

Mainland Computers
Leinster House
158 Leinster Road
Merivale
CHRISTCHURCH
Phone: (03) 554-339

Infodata Systems
Sixth Floor, BNZ House
Cathedral Square
CHRISTCHURCH
Phone: (03) 796-480

Ken France Electronics Ltd
40 Tarbet Street
PO Box 213
ALEXANDRA
Phone: (0294) 8021

Generation Associates Ltd
320 Princes Street
PO Box 508
DUNEDIN
Phone: (024) 770-126

inside BITS & BYTES . .

PLUS:

EDITORIAL

Bits and pieces

After six issues (you are reading the seventh) it is time to answer some queries (and dare we say complaints) from readers about the way we have done things.

First off, the packaging of the magazine. That changed last month with a switch to plastic bags which we hope improved the condition the magazine arrived in (no more ironing out the creases).

Second, back copies. We ask for patience with these. The demand was such that our copies of issue number 2 disappeared quickly. Because of that demand a special reprint (minus the covers) of 500 copies of issues 1 and 2 was done and most people should now have the back copies requested (sorry about the cover but the cost would have been too great). This is the last time a reprint will be done so if you want either of these two issues please hurry with your orders.

We also request new subscribers to be patient. It is too time consuming to process subscriptions individually as they arrive and recent demand has been such that our computer has been kept very busy. So all subscriptions start the month after they are received and if you wish back copies please use the order form provided.

Reader contributions. Keep sending them in but please don't expect them to be printed the next month. If you article or program is not acceptable we will return it otherwise we intend to publish it but that might not occur for several months so keep looking.

Believe it or not we've also had complaints from some readers that the magazine's price is too low. Reluctantly economics says we have to agree and the price will increase when the price freeze ends. So if you are not subscribing be in now.

Coming up in BITS & BYTES

Business Computing

The Calc-a-likes — just what are financial modelling programs and how can they help your business.

The first in a series by Peter Brown which will include a look at the popular programs such as Visicalc.

NZ business software. John Vargo reviews Auckland based International Applications Ltd "Charter" suite of business programs. Case study of a small business using a microcomputer for accounting applications.

Hardware Review

A new release from National Panasonic of Japan — the low priced JR100.

Alternative machine code programming — John Durham puts the case for using machine code monitors.

Plus columns on:

Farming
Education
Beginners

BITS & BYTES is published monthly, except January, by Neill Birss, Dion Crooks and Paul Crooks. Editorial and subscriber inquiries to Post Office Box 827, Christchurch.

ADVERTISING:

Auckland — Wendy Whitehead. Phone 794-807
Wellington — Annie Carrad: Phone 723-431
South Island — Graham Beecroft: Phone 554-265 Chch or telephone Christchurch 66-566

EDITORIAL:
Auckland — Cathy and Selwyn Arrow: 30A Bracken Ave, Takapuna
Christchurch — P.O. Box 827 or Phone 66-566
Wellington — Shayne Doyle, 280-333 ext 892 (W), 278-545 (H)

Production: Roger Browning, Graeme Patterson, Lee Teck Fui, Janine Morrell
Typesetting: Focal Point
Cover and Graphics: Sally Williams

Technical Editor: Chris O'Donoghue

Subscription rate: $8 a year (11 issues) adults
$6 a year for school pupils
Subscriptions begin from the next issue of BITS & BYTES after the subscription is received.
Back copies are available (except issue 1) at $1 per copy plus 50c per copy postage and packaging.

Subscription addresses:
When sending in subscriptions please include postal zones for the cities. If your label is incorrectly addressed please send it to us with the correction marked.

Distribution inquiries: Bookshops — Gordon and Gotch, Ltd. Computer stores — direct to the publishers.

Printed in Christchurch by D.N. Adams, Ltd.

mention BITS & BYTES when contacting advertisers

MICRO NEWS

IBM has reacted quickly to criticism and lowered the prices for the recently announced IBM PC.

The price for what IBM sees for a typical business configuration — 128K RAM, two 320K diskettes, keyboard, screen, and a dot matrix printer — has been cut from $10,932 to $9860, a reduction of 11 per cent.

As well this is post-devaluation i.e. the recent devaluation won't affect the price, meaning an effective reduction of some 20 per cent.

And those people who have already bought IBM PCs won't be disadvantaged. The price reduction has been backdated to the release of the PC in New Zealand.

IBM has also announced the release of 10 and 20 megabyte hard disks. The sample business configuration with a 10 megabyte disk replacing one diskette will cost $14,600.

★ ★ ★ ★

A direct competitor for the Apple II is now in use in at least one New Zealand school and is expected to be on retail sale here soon.

Selwyn College, in Auckland, has purchased 15 Franklin Ace 1000s, a "99.9 per cent Apple compatible computer", according to the International Sales Manager for Franklin, Mr G.R. Treseder, who visited New Zealand last month.

Now the Educational Trading Society is offering the Ace 1000 at $1850 ($1890 with colour) which is believed to undercut the education price for the new Apple IIe.

Nevertheless it does have a numeric keypad, 15 Visicalc keys and a built-in fan. It will be interesting to see if a price differential occurs on the retail

Apple Computer, Inc, has already lost a copyright suit against the Franklin Computer Corporation in the United States.

"It is not a copy, it is a record player that plays Apple music," said Mr Treseder.

Some of the Ace 1000s advantages over the Apple II, such as 64K RAM and upper and lower case characters, have disappeared with the release of the Apple IIe.

From previous page

Review of Franklin Ace 1000 soon.

* * * *

The Auckland branch of the New Zealand Computer Society is to hold a seminar "Microcomputers for the Layperson" from 9.30am to 3.30pm on Saturday 23rd April at the Architectural Conference Centre, Auckland University. $2 admission will cover lectures, videos, and discussions. Also on display will be a selection of books and computers.

For details contact Ian Mitchell on 583-350.

* * * *

The B.B.C. computer-literacy project in the United Kingdom has not stopped at the end of "The Computer Programme" series.

A second series is now on the air in Britain called "Making the Most of the Micro" and more are planned.

A proposed future development is to transmit computer programs from the television studio to the B.B.C. Microcomputer in the home using spare capacity in the television signal (similar to Teletext transmissions).

* * * *

Wrightson NMA intends setting up small computer bureau operations at its branches, using Sord microcomputers. Farmers will be given access to typical financial applications — cash-flow projections, account analysis, the ecoromics of farm purchase and analysis of longer-term developments.

* * * *

Kellogg farm management courses on the use of on-farm computers are still running. Workshops are planned for May, July and November. For further information, contact Mr J. Callan, Rural Development and Extension Centre, Lincoln College, Canterbury.

Industry seminar

The use of programmable manufacturing equipment to enhance the profitability of small-batch production will be the subject of a two and a half day seminar in Palmerston North on May 23-25.

The seminar will address itself to equipment that "works for you" — cutting metal, producing drawings, handling components

Topics to be covered will include computer-aided design, computer-aided programming for N.C. machines; C.N.C. and N.C. machines; flexible automation, and robotics. The main speaker on robotics will be Dr Alex Holzer, an Australian research scientist.

Speakers from Cable-Price Engineering and Walker Scientific (representing Computer Vision) will also contribute.

Further information can be obtained from: the secretary, Department of Production Technology, Massey University, Palmerston North.

Hands on the IBM PC

By Chris O'Donoghue

The machine as tested had two five and a quarter floppy disk drives, 64K of memory and a monochrome display/parallel printer interface with green on black monitor and an eighty character per second parallel printer.

When turned on the IBM personal computer performs a self test of all system components.

Documentation

Documentation consisted of three manuals, a technical manual, a BASIC manual and a general information/DOS manual (although the technical manual is an extra).

In general documentation was complete, easy to understand, with simple examples. But I did find some inconsistencies. For example, in the BASIC manual on the subject of conversion from single precision variables to integer variables, it stated that BASIC **truncates** single precision variables to integer, later in the same section the manual stated that BASIC **rounds** single precision variables to integer (this is in fact what it does).

The documentation comes nicely packaged in ring binders each with a case. All documentation is fully indexed and easy to follow.

Hardware

The IBM PC is based on the Intel 8088, a chip with an internal 16 bit structure and an 8 bit data bus.

Inside the system unit is 64K of RAM, 40K of ROM (used for boot, system check and BASIC) a cassette interface, and room for disk handlers. There is also a keyboard interface.

Then comes the five expansion slots, I feel this is carrying the "simple system with user defined expansion" a bit too far.

There should at least be a display interface included with the primitive system. Instead one of the five expansion slots was used for a monochrome display/parallel printer interface costing $699, leaving four slots to plug-in expansion boards. Of these there are options of up to two memory boards (giving expansion up to 544K of RAM), a colour/graphics display board, various communications options.

Keyboard/Display

The keyboard has 83 keys, including 10 function key, numeric keypad, cursor movement and special keys such as a print screen key.

An 8 bit character set is easily accessable with the lower 7 bits giving the ASCII character set. The other 128 characters are graphics characters.

The keyboard is connected to the system unit by a spiral cable and typing angle can be adjusted.

This is a very good keyboard and it has a wonderful feel.

The display is green phosphor with 25 lines by 80 characters

wide.

Characters can be underlined, blinked, set in high intensity or reverse video. Full upper/lower case and graphics characters are displayed. Screen can also be set so that typed characters are not echoed (good for passwords etc).

Disk:

The disk drives are either 160K or 320K depending on how much money you have. They take 5¼ inch floppy diskettes and are not too noisy. Up to 2 drives are supported.

Software:

There is a large software base available for the IBM PC. These include operating systems such as CP/M-86 and UCSD p-system, compilers such as BASIC, PASCAL, FORTRAN and COBOL and a host of packages from VISICALC and MULTI-PLAN through EASY WRITER to the Business packages, Charter and Attache, and games.

Operating System:

The operating system PC-DOS will support one or two diskette drives. File structure is of the type DEVICE: FILENAME. EXTENSION where FILENAME is up to eight characters long and EXTENSION is up to three characters long.

No attempt seems to have been made to include a hierarchical structure such as sub-directories.

Apart from handling files the DOS does the other common things, displaying directories, renaming files, copying files etc. There is also a nice line based text editor called EDLIN which I found very useful in changing ASCII files.

BASICS:

BASIC on the IBM PC comes in a confusing four versions.

1. Cassette BASIC, included in ROM, is a full BASIC that handles cassette I/O, printer output, and light pen functions.
2. Disk BASIC, loaded from disk, it is a full superset of cassette BASIC. In addition it handles disk I/O, communications (if a communications interface is present). It also has time and date functions.
3. Advanced BASIC, is again a superset of Disk BASIC, but includes event trapping and colour/graphics function (if the colour/graphics board is present).
4. Compiler BASIC, an optional compiler for $709 (not tested).

Of these four versions I tested the three interpreter BASICs. These are all upwards compatible i.e. anything written in cassette BASIC will run on Disk and Advanced BASICs etc, but not so the Compiler BASIC.

I used the Advanced BASIC most of the time to get the widest possible range of functions. The BASICs are written by Microsoft and Advanced BASIC is a superset of standard Microsoft BASIC.

A nice feature was the event trapping which was available for function keys and communications events (e.g. receiving data). This was of the form:

ON event GOSUB n

In the case of function key event, trapping was KEY(m) where m is in the range 1 to 10.

Communications were also handled nicely. They were treated like a normal file i.e. after doing an OPEN "COM... statement normal PRINT # and INPUT # is all that is needed. No INP(n) etc are needed (if you have the right interface).

Overall IBM seems to have made a reasonably conservative product. The BASIC is not overly structured, the DOS doesn't handle complex directory structures etc. But the lack of major faults in this computer is remarkable. However with the amount of software available and sure to be available soon this machine should take a large share of the market.

This is despite the price tag (typical small business configuration around $10,000) but is partly because of the label which seems to be perhaps the biggest selling point.

Microcomputer Summary

Name:	IBM Personal Computer
Manufacturer:	IBM
Processor:	Intel 8088
Clock Speed:	4 MHz
RAM:	64K-544K
ROM:	40K
Input/Output:	Two 5¼ inch floppy disk drives, cassette interface, Centronics parallel printer port, Video, 5 expansion slots.
Keyboard:	83 key, auto repeat, numeric key-pad, editing keys, 10 programmable function keys.
Display:	25 lines by 80 characters, upper/lower case, underlining, blinking, reverse video, green phosphor.
Languages:	Microsoft BASICs (see text) compilers for Pascal, BASIC, Cobol, Fortran, Macro-Assembler.
Disk Operating Systems:	PC-DOS, also CP/M 86, UCSD, CP/M 80 softcard.
Graphics:	Character level graphics on system tested. With additional Colour/Graphics board: 320 x 200 4 colour, or 640 x 200 2 colour.
Sound:	Sound generator and speaker.
Cost:	System as tested approx $9,000. Additional: colour/graphics card — $658, Asynchronous Communications Card — $313. These prices subject to price reduction announced late March.
Options:	Many Hardware and Software options exist. Including memory expansion to 544K, SDLC communications.
Peripherals:	Printer, disks, other machines (e.g. laboratory machines) through communications adaptors.

IT'S HERE! the IBM of Personal Computers.

For 40 years IBM has built up a wealth of experience and expertise in computers. Now that knowledge is built into a tool for modern times: the IBM Personal Computer.

As with any new tool, you'll want to get comfortable with the IBM Personal Computer before putting it to serious use. You'll have some step-by-step reading, but our instructional literature involves you from the start. And the Computer is on your side too — interacting with you as you learn. There's no reason why you can't be executing programs and feeling good with the results within your first week. After a month, it should be clear that you've made a good investment, and you'll probably be telling your friends why they should get one.

To keep up with modern times, visit your authorised IBM Personal Computer dealer today — or call at the IBM Product Centre in Auckland, Wellington or Christchurch.
Ask what programs are available now. Get a demonstration.
Be sure you check out the specifications and features that make this the IBM of Personal Computers — features that set it apart.
You'll find that the quality, power — and realiability — are what you'd expect from IBM. The price isn't.

IBM

HELPING PEOPLE FIND THE ANSWERS

The IBM Personal Computer.
A tool for modern times.

Whangarei: M.M. Baigent & Co. Ltd. Phone 84-979. **Auckland:** Byte Shop Phone 32-860. Computerland (NZ) Ltd. Phone 798-005 or 798-078. Financial Systems Limited Phone 789-068 or 789-069. International Office Equipment Phone 775-372. **Hamilton:** Thomson & Ward (1971) Ltd. Phone 82-679. **Rotorua:** Thomson & Ward (1971) Ltd. Phone 479-172. **Tauranga:** Bay of Plenty Office Supplies Phone 81-009. **Hastings:** Midland Data Processing Phone 84-528. **Palmerston North:** Barlow Electronics Ltd. Phone 70-845 or 70-849. **Wellington:** Compusales Phone 844-146. Project Computers Phone 731-152. Wordcom Bureau Phone 729-028. **Nelson:** Geo. Berryman Ltd. Phone 81-489. **Christchurch:** Small Business Software Ltd. Phone 64-617 or 64-717. **Dunedin:** Whitcoulls Ltd. Phone 774-120. **Invercargill:** Office Equipment Southland Phone 84-448. * IBM Product Centres. **Auckland:** Phone 778-910. **Wellington:** Phone 729-499. **Christchurch:** Phone 792-840. **Suva, Fiji:** Kelton Marketing Ltd. Phone 385-533.

NEC PC 8000 — "effective productivity tool"

By Shayne Doyle

One of the five computers recommended for use in schools, the NEC PC-8000 system loaned to me for this review was a CP/M colour business version comprising:

- PC-8001B CPU/Keyboard console
- PC-8011B Memory expansion & Interface unit
- PC-8031B Dual floppy disk drive
- PC-8023B Graphics printer
- PC-8043B Hi-Res colour monitor.

Price, around $8,600.

Over the week or so that I had the machine, I grew to like it very much, parting with it reluctantly. I found it extremely easy to use, in fact, within the first two hours I had started writing this article using the CP/M Word Procesor SELECT.

Physically, the system is similar in size to most other business computers in it's class, but I was most impressed by the solid well engineered system components. The CPU/Keyboard unit is very definitely designed to withstand hard knocks and a lot of use, being a substantial metal enclosure. There is very little use of lightweight plastic anywhere in this computer system.

Unpacking and connecting the system together took me a half hour and the only problem I had was with the cable between the disks and the expansion unit — these are edge connector types and if not pressed firmly home they can grip the edges of the circuit board without mating properly with the foil tracks. Apart from this, the whole system fired up and proceeded to load disk Basic as soon as it was switched on. Normally, however, this system would be installed for you by TISCO under contract to Scollays.

By stacking the disks, expansion unit, and monitor behind the keyboard (remembering to preserve ventilation airflow space), and sitting the printer alongside, a very compact installation can be achieved. This also situates the monitor at a comfortable eye level.

The do-it-yourself programmer will enjoy working with the NEC's Basic screen editor — I found it very versatile, and it made writing code so much quicker than with a line editor. I also found I preferred using the 40 column screen width, with the consequent larger character size, when programming. The display may be easily reformatted with the WIDTH statement, and CONSOLE sets up other display feaures.

The standard graphics facilities are not particularly impressive, with only 160 x 100 pixel resolution, a LINE statement, and what are effectively SET and RESET statements. I did, however, have a disk of 12 machine code

Microcomputer Summary

Name:	NEC PC-8001
Microprocessor:	uPD780C-1 (Z80A equivalent)
Clock Speed:	4 Mhz
RAM:	32 K Bytes Dynamic
ROM:	24 K Bytes for Basic and machine language Monitor
Input/Output:	Colour and B&W video 600 baud FSK cassette Parallel Printer interface I/O Expansion Interface
Keyboard:	Full ASCII with numeric keypad, 5 shifted programmable function keys, alternate character set, graphics set from special graphics "shift" key.
Display:	Format user definable — 20/25 lines per screen, 36/40/72/80 characters per line.
Languages:	NEC N-Basic, Z80 machine code assembler. Fortran, Cobol, PL/Z, Pascal, Lisp, Basic Compiler.
Graphics:	128 graphics characters, 160 x 100 pixel resolution, 8 colours, variable intensities.
Sound:	Built in speaker.
Cost:	Basic unit cost — $2140 which includes 32K CPU keyboard console and Green screen monitor.
Options:	3 different expansion units for memory expansion, adding floppy or hard disk drives, parallel/serial/IEEE488 I/O ports, extra printer ports, priority interrupt real-time clock, slots for user PCB's (printed circuit boards), Light pens, Voice recognition units, Flat Bed Plotters, Hi-Res graphics (640 x 200 pixels).
DOS Options:	NEC DOS, CP/M, UCSD p-system, RACET DOS.
Peripherals:	PC-8043 Hi-Res Colour Monitor $1050 Medium Res Colour Monitor $470 PC-8023 Dot Matrix printer $1440 PC-8011 Expansion Unit $1210 Minimum disk interface $250 PC-8031 Dual mini disk drive $2340 CORVUS Hard Disk Drives up to 20 Bytes.
Other Features:	Terminal Mode — allows the computer to operate as a standard terminal with emulator software for most IBM and other mainframes.

graphics games, which demonstrated that quite effective dynamic displays can be achieved.

The LINE statement has parameters which control colours or, if black and white mode, video attributes such as reverse, blinking, secret, and combinations of these. The graphics and alternative special character sets, while residing in the usual decimal 128-255 positions, can also be entered directly by a special graphics shift key, and alternative character set shift key.

Extensive print formatting facilities using "mask" techniques are available, and the same formatting is used for writing basic sequential disk file records. Both sequential and random access disk files are handled under NEC DOS Basic, and it is quite literally "child's play" using the disk features.

A very extensive repertoire of functions is offered, more than is usual, with quite a few devoted to conversion between the various variable types. There are functions to convert between decimal, octal, and hexadecimal, and in fact, the latter two bases may be used directly by using special prefixes to the numbers.

Both date and time functions interrogate the computer's internal clock. Programmers used to TRS-80 or System 80 will be glad to see the very useful VARPTR variable pointer function.

For the "die-hard" hex machine code programmer, a basic machine language monitor is provided, offering eight commands — display and alter memory, load and write tape, verify tape, go to address, test memory, and return to Basic.

Moving away from DIY software, adding the RACET disk operating system ($250), allows the use of keyed files, sort utilities, etc. A Spreadsheet program ($104), and a Word Processor ($130) are available.

Expanding to a full CP/M Disk Operating System ($200), gives access to the vast range of applications software written for CP/M computers. These include accounting systems (PADMEDE, IAL, GOLDFINCH etc), Budgeting

and Planning programs (Report Manager, Supercalc, Scratchpad, T/Maker etc), Database management software (DBASE II, FMS-80, CONDOR etc), Word Processor programs (SELECT, BENCHMARK, WORDSTAR etc).

For more sophisticated custom written programs, a choice of language compilers — Basic, Pascal, Cobol, etc.

Documentation was very well covered, a comprehensive manual being provided with each component of the system, NEC Basic reference manual and quick reference programmers card. The applications programs had their own detailed manuals.

The quality of the manuals is excellent, and the only adverse comment I would make is on the lack of a detailed index in the back of the manuals, although there is a table of contents at the front of each.

Apart from this small point, I could only really criticise this computer on three counts.

Firstly, the printer is rather noisy in use and would benefit from additional sound proofing. It also emitted an annoying whistle (about 9-12 Khz) when turned on. Enquiries established that this peculiarity only appears in printers manufactured up to a certain serial number. While this printer is bidirectional, it was tedious to find that SELECT does not take advantage of this, and only output print in one direction.

My second gripe is minor, and reflects the basic nature of the standard graphics facilities. There appears to be a limit to the number of pixels that can be SET at any time, and I also felt that some form of programmable character generator should have been available, not to mention some sort of "sprite" system.

The third point is of interest to the computer hobbyist — this machine does not have a parallel or serial port on the basic console, these are in the expansion units.

The business user will not, however, bother about these latter two criticisms, and in my opinion could consider this computer a very cost effective productivity tool. As you may gather, my overall reaction is definitely favourable, and I enjoyed working with the NEC PC-8000 computer system.

A clever disk-drive expansion for the System-80

By JAY D. MANN

Until now, System-80 owners wishing to expand their computers have had to purchase one or more adapter boxes to be added via the expansion connector at the back of their keyboard unit. They can now purchase a pair of locally made units that provide expansion to full memory plus disk and printer operation in an elegant fashion.

There are two parts to the expansion. First 48K of RAM is installed inside the System-80 keyboard box. Second, a combined disk drive/disk controller box is plugged into the expansion connector in the back of the System-80. (You could use a disk drive without any memory expansion, as I did briefly for this review, but the software involved in controlling disk operations is so bulky that there won't be enough room in your memory for many useful programs.)

The in-keyboard expansion is done by the developers, Micro Processor Services of Christchurch, or by one of their agents. This step should not put you off, for it is an excellent way to expand memory. They have local agents throughout the country. Alternatively, you could mail them either one board from your System-80, or the entire computer. If an external expansion interface is used, it contains a great deal of circuitry that duplicates logic, buffering, and timing circuitry already present inside the keyboard unit. By using the in-keyboard memory expansion, there is no need to duplicate these circuits. You have a computer that has a full complement of memory without any expansion board, adapters, and extra cables.

In addition, if you install a clock speed-up modification, the in-keyboard memory has no trouble keeping up with the increased throughput, whereas many expansion interfaces have internal timing circuits rigidly locked to the standard System-80/TRS-80 clock. Such expansion units cannot work properly with computers working at speeds.

Wait a moment, you must be saying. What about the other functions of an expansion interface, such as a disk drive controller and a parallel printer port? The good news is that the folks at Micro Processor Services, Ltd, have come up with a compact, well-designed expansion board that is built directly into the base of your first disk drive. No additional boxes or cabinets are needed. A single 50-pin parallel cable connects the System-80 to the disk drive/controller. The unit provides standard parallel cable connectors for additonal drives, a printer, and further use of the System-80 bus. Think of it. A single cable between your computer keyboard and the disk drive, and you have a complete system that is still highly portable.

The expansion board is not a bare-bones design but instead provides a number of refinements not available on expansion boards designed overseas. These are controlled by a row of toggle switches set underneath the disk-drive door.

The most useful addition is the ability to fool the computer into thinking that a printer is attached when it is not. This can be very important for programs that normally LPRINT at some point. If you have just spent a long session entering data, and then the program jumps to an LPRINT statement, normal System-80 computers will lock up unless a printer is actually plugged in. Pressing the Break key won't help; you have to push the Reset button, and lose all you data. The Printer Disable switch bypasses the LPRINTing. Your results won't be printed (obviously, if you've no printer!), but at least the program will continue to run. You could, of course, edit the entire program to change LPRINT to PRINT everywhere, but one slip-up and all is lost. (Some but not all disk-operating systems let you route printer output back to the screen. Again, this works well if you remember to do so before and not after the computer gets hung up.)

If you do have a printer, then the expansion unit provides a very well designed parallel printer port. Not only does it respond in the System-80 "OUT FD" instruction but also to the TRS-80 "LD (37E8), A" instruction as well. No switches need be thrown. This overcomes one of the major difficulties with the "improved" System-80 software using a proper Z80 "OUT" instruction . . it is incompatible with a large amount of TRS-80 software that tries to use the 37E8 approach.

All would be well if all printer commands went through the ROM routines, but because the ROM routines are rather minimal many advanced programs are designed to control the printer directly. Thus, the New Zealand-designed board solves the problem neatly by responding equally well to System-80 and to TRS-80 printer commands.

Two separate reset switches are provided. One is like the normal reset button on the back of the computer. The other one acts like a power-off reset. A particularly useful switch allows you to fool the computer into thinking it is a non-disk system. An annoying aspect of disk-expanded System-80 and TRS-80 computers is that hitting the Reset button fills the screen with garbage while the computer waits for the operating system to be loaded off the disk — even if you don't have a disk in the drive or are operating without a drive connected.

A Write-Protect switch blocks transfer of data to a disk, so that you can do certain risky operations without putting garbage on your disk.

Finally, room has been left for future addition of a double-density controller to the system. A switch will then determine whether the system boots up in single or in double density. The System-80 normally starts up in single-density. If suitable hardware is present, it can then be switched to double-density.

On examination of the interior, the disk controller unit can be seen to consist of a separate

power supply plus two well-planned double-sided printed circuit boards, with plenty of room and no likelihood of heat build-up. The two boards are stacked one above the other, and fitted below the MPI disk drive. The boards are linked to each other by a 20-pin plug-and-socket connector that provides some of the mechanical support for the upper board, and will simplify any necessary servicing. A flat cable leads upwards internally into the attached disk drive unit, which has its own independent power supply. Typical of the thought that has gone into this unit is the mounting of the U-shaped disk-drive cover. In addition to the customary four screws, there is an additional pair of screws that serve as hinges so that the cover can be swung up out of the way. (For those who may be confused about terminology, a disk is the 5-inch flat object that spins around inside a disk drive. The disk drive itself is pretty unintelligent, so the job of getting information on and off the disk in a form meaningful to the computer is done by the disk controller.

You may have gathered that I'm enthusiastic about this system. I am. My present configuration is a System-80 with a separate 50/40 adapter that in turn leads to an LNW expansion board in its own big case. From this, one cable snakes out to the disk drive and another to the printer. When I tried out the MPS configuration, with the combined disk drive/controller box sitting on my table next to my keyboard unit, the extra desk space and lack of clutter was most impressive. The operation of the controller was, as it should be, unobtrusive. Single-density disks booted reliably and the printer responded to LPRINT and LLIST commands. When I disconnected the printer and threw the Printer-Deselect switch, I was able to run programs that called for LPRINTs without the computer locking up.

This combination of in-keyboard memory expansion plus a disk controller built right into the first disk drive, is not available anywhere else in the world. It should have a market not only in New Zealand but overseas.

SOFTWARE

Learning in space

Apple Software for schools: Multiploy. Available for Apple II 48K on disk. Retail $36.95. Reviewed by Kathy Broadley.

The microcomputer as an educational tool has not been exploited yet in New Zealand. Few primary schools have even one micro, so the use of this potentially useful teaching aid is just a gleam in the eye of some teacher enthusiasts and some hopeful parents. Perhaps this is just as well since many of the so-called educational packages are technically competent but educationally doubtful. The objective of getting the user to practise number facts can be achieved with a spirit duplicator worksheet for a fraction of the cost of a computer program, even though the immediate reinforcement of knowing whether the answers are correct is probably handled best by the computer

Because I have seen some poor packages I was somewhat sceptical when I loaded "Multiploy", by Paul Coletta (Reston Software, Prentice-Hall, distributed in New Zealand by Whitehall Books). With this package I was pleasantly surprised. The accompanying booklet claims that "Multiploy" "combines the excitement of an arcade game with the challenge of learning and practising arithmetic skills." The target age range is four to 14 years, "but adults like it, too". There are three levels of difficulty and the user chooses from the four arithmetic operations. Within each level the speed at which the problems must be answered can be manipulated. A nice feature for classroom use is the option of low-level sound.

The arcade game aspect is catered for in the way each problem is presented. Each problem appears in a smiling (space)ship. If the correct answer is entered the ship is shot down. If the problem ship is not destroyed, it will eventually start shooting back, possibly destroying the user's answer 'base'. Up to four problems are on the screen at once, descending towards the 'base'. The user is given a rank that is determined by the number of correct responses. A perfect score at any level attains the rank Multiploy.

For classroom use a recordkeeping feature would have enhanced the program's value. A record of problems incorrectly answered by each user could be of assistance to a teacher making decisions about what should be taught next. Equally useful would be an authorising system so a teacher could insert his/her own problems. Perhaps future revisions of "Multiploy" might include these features. I also wondered whether more traditionally-known ranks such as Lieutenant or Admiral might be more attractive to children.

The instruction booklet is clear enough for a fairly amateur operator to handle. That's important since most teachers will fall into that category until micros are more commonly used in schools. More information on the scope of the problems included at each level would be useful.

This arithmetic practice program uses educational aspects of the microcomputer intelligently. The motivation of the arcade game format is harnessed. It is a drill and practice package, not a teaching one. As a drill and practice package it is more imaginative and lively than most, and would be useful at all levels of the

primary or intermediate school. The harshest critics are the consumers, so I tried "Multiploy" on my two 13-year-olds and 10 and nine-year-old friends. It emerged with flying colours.

Games for the VIC

VIC-20 Games. Tapes 1 and 2 James Electronics Ltd, Box 527, Thames. $15 each. Reviewed by A.J. Petre

With three good games a tape, the James tapes are good value. They also make excellent use of graphics and of colour — too many programs these days have badly-chosen colours that do not contrast. The result is illegible lettering and invisible games symbols.

TAPE 1 has "Snake" "Maths Game", and "Ball in Bucket". The first two need memory expansion which should be shown more obviously on the packaging. "Snake" is similar to the caterpillar-style of game — your snake (which will screen-wrap) gets longer as it gobbles mice. After a certain time mice turn into gravestones, and if your snake bites them, or itself, it dies. Good sound, quite addictive.

"Maths sets the task of making mathematical expressions of varying difficulty equate to an object number — a timed game against the VIC. Good stuff for making maths fun for the kids, and hard enough in its upper skill-rates to keep even the whizz-kids guessing. A good mind stretcher.

"Ball in Bucket" is a simple but good-fun dexterity game, in which the player uses specific symbols to bounce a ball into containers.

TAPE 2 needs no expansion, and has "Formula 1", "Concentration," and "Line Game". Two of these are for two players — something we need more of in computer games.

"Formula 1" is a fairly typical race game, with good graphics and colour, but only average sound. You have to get around the track without running or spinning off, wearing out brakes and tyres. The car is hard to see at times which leads to frustration and crashes. That apart, quite a good example of its type, but less gripping than some.

"Concentration" s simple, but addictive. Two players seek mystery symbols under the letters on the screen, the aim being to pair them up. It needs a good player memory and good attention — with good colour and graphics, it's a winner.

"Line Game" is a simplified version of the "Snake" or caterpillar game — steer two lines (one player each) into or away from collision. Simple but fun.

One point: "Formula 1" and "Concentration" have good written instructions on the leaflet, but none in the program. A bad idea, I think. We all end up losing those little bits of paper, then it's hell working out which keys to use.

My rating for both tapes:

Loading and instructions: Good.

Colour and graphics: Very good

Value: Very good.

Player interest: Good to high.

A final point: these are streets ahead of many imported games... bet they'd sell elsewhere. Governments, please note.

LETTERS

Taxing the computer

Sirs — Many thanks for a very informative article on duty and sales tax on software etc (February '83 issue). As I write this, I still await an answer to an enquiry from the custom house in Wellington. Perhaps, the computer they use can't or is still computing the variety of tariffs available to the department.

It is a pity I didn't receive this issue until the day of "media for data processing equipment" or they may have had this jumble of words.

Once again, thank you and keep up your informative reporting.

I.McDONALD (Taupo).

It's less hassle regularly

Having difficulty getting your copy of "Bits & Bytes" regularly? We suggest that you become a subscriber. If you prefer to buy from your bookstore or computer shop place a regular order. This will help ensure you always get a copy.

VIC03

COMMODORE VIC 20 HOME COMPUTER— ONE OF THE FAMILY!

World renowned VIC 20 — Commodore's brilliant home computer is at your dealer's today! It's a family machine — with something for everyone . . . play the best video games available . . . or teach yourself a language . . . write your own programmes . . . even use it to budget for that new car! (And that's just a tiny sample of what it can do!)

VIC 20 plugs straight into your television set and comes complete with

- colour and sound
- full size typewriter keyboard
- graphics capability
- easy to learn BASIC language
- easy to follow instruction manual
- full service and support

Take a look for yourself! The affordable Commodore VIC 20 is at your dealer's today

commodore COMPUTER

End of the series

Computer graphics is now recognised as a major form of computer addiction. A great deal of thought goes into deciding just what a particular microcomputer will offer you for graphics . . . . and a good deal more thought goes into designing tempting add-on devices that you can buy at a later date. Such devices can help you enormously in creating, displaying or interacting with graphics. Some are cheap and some are expensive.

This last article in this series outlines some of the clip-on graphics devices available on small computers. Basically such devices can perform one or more of four functions. They can help you create graphics more easily; they can help you display better graphics and finally they can help you process graphics more effectively.

The clip-on path to better graphics

By PIP FORER

The most common graphics add-on that can be found is the paddle, often called the games paddle. The Apple II at one time came complete with a pair. Today many people purchase them at an early stage for an easy entry into the world of computer games. The paddle is usually a very simple analogue device with a rotating knob (connected to a potentiometer) and a simple on/off fire button. It relies on the user's software for interaction with the screen.

All the knob on the paddle does is return a value in a certain range (typically 0 to 255) which is determined by the position of the potentiometer. Statements in BASIC or machine code translate this value into a screen position for drawing. The user's eye then appreciates this position on the screen and adjusts the paddle appropriately to affect the image he is getting. A single paddle is usually used to allow the user to control an object moving in one direction, say the laser turret in Space Invaders. Two paddles allow control in two dimensions.

More seriously, paddles are often used in screen creation utilities; for instance, to position a cursor for the start of a label or for locating a shape to be drawn.

A more sophisticated version of the paddle is a type of joystick where movement of a joystick can capture two potentiometer values at once, one for X and one for Y co-ordinates. The principle is identical to having two paddles but is made mechanically easier for a human hand to cope with. Most Apple joysticks are of this type.

A second type of joystick is one where a direction rather than a position is sensed. This is typical of Atari joysticks, the most publicised of which is "Le Stick", a free-standing column in which direction is sensed from the simple angle of the hand holding it. Joysticks such as these return codes corresponding to directions of tilt (up, down, left, left and up etc), usually returning nine possible states. These correspond to stationary and the eight sectors of the compass (N, NE, E, SE, and so on). Again the link to a program consists of sensing the codes returned by the joystick and acting accordingly. In the case of controlling the movement of a shape on the graphics screen you would need to check the joystick response. If it was "move up and left" then the X and Y values of whatever you were drawing to the screen would have to be adjusted to do this.

The main operational difference between the two types is that the latter sort requires you to check that your object stays on the

screen while the former ones can be scaled to ensure that the screen boundaries are always conformed to.

Joysticks and paddles come in various qualities (in terms of both ruggedness and purity of response) and are fairly cheap. Other means of screen interaction are more expensive. The most popular is the light pen. Light pens can be used to simply point at a display screen so that the computer records where the pen is pointing. This is a bit more direct than piloting a cursor around with paddles. It differs from paddles and joysticks by also involving the screen directly. Most decent light pens sense the location of the pen by very sensitive timing. The screen on a monitor is refreshed regularly by an electron beam and a sensor in cheap light pens actually picks up the passage of the refresh beam over the screen. By comparing the time at which this is sensed with its knowledge of where the refresh from the CRT should be at that point in time it can calculate the X and Y position on the screen.

A word of caution. Because of timing rates that differ American light pens give peculiar results on New Zealand machines using standard monitors. This is because the refresh rates vary in certain countries. Also the poorer pens only give a limited standard of accuracy. Machines with built-in monitors (such as most of the Japanese machines) are able to offer a far higher accuracy of response than machines with ad-hoc display units. Many 8-bit and most 16-bit machines now offer a light-pen port.

Finally there are mice. A mouse is a small, hand-held device which can be moved by the hand over any flat surface and translates its own movements into a cursor movement on the screen. The mouse achieved a small fame with Xerox, which used it for screen interaction with a language developed at its Palo Alto research centre and called Smalltalk. It has recently found wider publicity as the main interactive device employed by the new 32-bit Apple Lisa computer. The computer press have applauded it. Others are imitating it but you will need almost $US10,000 (and I suspect some patience) to get within strike of one just at this moment.

Capturing graphics

The second sort of peripheral is aimed at capturing an existing pattern on the computer. The two most common clip-ons in this field are bit pads (alias graphics tablets, alias digitisers) and video-digitisers (alias frame grabbers). These are typically quite expensive peripherals but extremely time saving.

Bit-pads are the most common graphics peripheral after paddles. The basic aim of these devices is to allow the user to draw indirectly on the screen through using a hand-held pen or cross-hair cursor. In general they are used where an existing line-drawing exists (say an outline of New Zealand, a cartoon character or a diagram) and the pen can trace the design from a table top and see it appear on the screen. This has great attraction for the 99 per cent of computer users with limited artistic talent and/or a dislike of coding long strings of co-ordinates. Using such a device a complex shape, such as the North Island, can be encoded to the screen in a few minutes. The information coming in from the pen can also be used for other tasks, such as calculating the area of a shape by drawing around its exterior. Geography students use this regularly to calculate the area of water catchments for instance. The digitiser really is a great boon.

Unfortunately good ones are expensive. If you digitise information or shapes from large Lands and Survey map sheets you need both a large digitiser (able to deal with 30 in. by 40 in. maps) with very high accuracy (down to 1/100 in.) That is a five-figure piece of equipment. The high quality, smaller models weigh in from $1500 upwards. These usually consist of a flat tablet perhaps 30 cm. square which is connected to the computer by a cable. A pen, also connected to the computer, allows you to trace

lines. The principle of operation involves a fine mesh of wires embedded in the tablet and more fine timing. Essentially readings from the wire mesh and the pen tip are compared and the timing of signals allows the position of the pen tip to be computed.

These digitisers are accurate but expensive. For many microcomputers, whose owners simply want to draw a picture on the screen, the accuracy (and cost) is unwarranted. A cheaper set of digitisers exists using a hinged arm to calculate position. The hinges contain potentiometers (useful things that they are) and as the hinge is opened or closed the angle (and setting) of the potentiometer changes. From this the position of the cross-hair can be calculated and an image traced from a table top to the screen. Such digitisers need no special table but still serve only a limited size dictated by the length of their arms. An American firm (Penguin Software) has taken the whole idea one step further by having such a device that records 3-D shapes. It markets this with software that displays such shapes straight on the screen in perspective from any angle.

Far less common but an area of increasing interest are the video digitisers. These simply take the input from a television camera and convert this into a computer graphics picture. The incoming signal is broken down into a particular grid of pixels (say 256 by 256) and stored in graphics RAM just like any other picture. From here you can superimpose on it, clear bits out or modify it. The capture is fast, sometimes one-fiftieth of a second, and the image often good. These devices are the basis of "computer photography" side-shows. Until recently most cheap video digitisers needed American standard NTSC input so a New Zealand user might need a special camera for the task. British manufacturers have begun to produce systems based around PAL which may offer a better option for us. If you own a camera already (a big if) video digitisers are about the cost of a low-grade bit pad.

Better graphics outputs

Graphics is essentially about display, either hard copy or soft. With soft (screen-based) output you can enhance your image by essentially buying a better monitor. The problem is that a law of diminishing returns per dollar spent sets in. Rough video is cheap: you use the television. If you are dissatisfied with that a proper colour monitor of the large-volume production kind is relatively cheap. If you go for a basic RGB monitor it should cost you less than a home television. It gives a calmer picture.

Incidentally, RGB stands for red, green, blue, the three colours of the individual colour guns in the set. RGB monitors are quite nice since on some machines you can interface them to the computer through software-programmable cards. These can be programmed to set a background colour (set one gun on all the time), delete a primary colour (disable a gun) or give coloured text on a machine that does not have such a facility inbuilt. However, most low-cost monitors (and certainly all televisions) have an upper limit to the number of dots they can resolve. On most it is under 300 points vertically. To improve on this requires special circuitry (which is higher cost and produced in smaller volumes). These monitors employ a technique called interlacing to extend the resolution.

Thus the ACT Sirius gives 400 by 800 points monochrome while the astonishing NEC Advanced Personal Computer (not to be confused with the PC8000) offers 800 by 640 colour on its built-in monitor. Resolution above this gets progressively more expensive.

The other peripheral is the printer. Black and white graphic printers are now commonplace. The great growth recently has been in colour printers. Although exotic ink-jet ones exist in the over $7000 price range, matrix-dot printers offer colour at under half this price. Their operation is through a ribbon with at least three horizontal coloured stripes in it. To create a coloured screen merely requires the printer to pass several times over with the required mix of the primary colours being struck. A screen dump can take three minutes with this but the effect can be striking and cheap.

Putting punch in the processing

The final graphics option is to upgrade your processing power. This can be done with one of several options. The most common is to firstly increase the size of your memory. This just allows more graphics screens to be stored in the computer and rapidly accessed. The next option is to increase the speed of your processor. This means (for those machines for which it is available) possibly a second processor that is faster or one designed specifically for processing 3-dimensional data bases. Such add-ons are available for certain machines. Finally you can do a bit of both. It seems fair to end up with the Apple II for an example of this. If you are constrained by normal Apple graphics then roughly $1400 will get you a board with RAM, a graphics processor and 512 by 512 picture resolution. That might be one way to stick with a familiar system but pursue new heights. You will have to get a better monitor to cope with it though.

The main lesson from this discussion, and the preceding articles, is the complexity of any computer system, even a microcomputer. The use of graphics involves a whole spectrum (acknowledgements to Clive Sinclair) of techniques and equipment and all must be well matched. Good programs, fast processors, friendly interaction, good resolution are only part of the whole. There is little point having a great computer and a bad monitor or a super program but a weak processor. If you want to get into this area, or are already in it, mix and match intelligently to build a system that fits your budget, uses all its parts fully and does the job you want done.

Getting at the beast

by CHRIS McLEOD

Should you be using a computer on your farm, and if so, in what ways can you get access to a computer.

There are three reasons for farmers using computers on the farm:

- If you are interested in computers and their application on farms, you may want to use one as a hobby.

Any benefits you gain could be considered a bonus; they are not essential in fulfilling your aim. As long as you get personal satisfaction and can afford to use the computer, then that is reason enough.

- If you would like to improve some aspect of your farming operation, but not necessarily improve your financial position (eg. improve per head stock performance).

A computer could help but you must carefully evaluate the software available to ensure you can use a computer to achieve your aim. It would be best to seek advice from someone familiar with both farming and the computer software available. Again, you must be able to afford to use a computer.

- If you want to increase the profit from the farm, a computer could be used to help increase income and/or decrease expenditure.

This is the most likely reason, and involves the most work in evaluating software, costs and benefits. Unless you have a good knowledge of what software is available, how you can use it, and how you can put a dollar value on the benefits, I strongly suggest you seek advice from someone who does.

Farmers who already own computers, farm consultants, and computer consultants would be the best people to talk to (but remember some farm consultants may not be familiar with computers).

Idea — Before you commit yourself, you must have an idea of the annual cost of using the computer (including any extra time needed to collect and enter the required data), and the annual financial benefits. Some costs and benefits are quite obscure. For example, what value do you give to better management information which will allow you to make better decisions.

Now we will look at the various ways you could get access to a computer.

There are several ways you can make use of a computer without using it yourself. This could be through farm consultants; accountants; farm secretaries working for consultants; accountants etc; stock firms; dairy companies; data bank accounting service; and bureaus.

Several farm consultants are now using computers to process information from clients' farms. They can more easily derive useful management information from financial information such as you would provide for your accountant. They can use this to advise their client, or give the information to the client for his own use.

Because of the amount of work involved in maintaining physical records, most consultants would restrict the use of their computers to financial information. The advantage of using a computer in this fashion is that the cost is low, and the consultants can provide a great deal of information and help which a computer could not.

Most accountants now use computers to process their

clients' data, but very few provide anything more than a set of accounts – useful for the tax man, but not ideal for management purposes. Much of the information which could be used for management decisions is there, but not in a form which can be easily interpreted.

At relatively little extra cost, accountants could provide farmers with a considerable amount of information which would be useful for management purposes. We hope to see more of this in future.

If farm secretaries were to carry a computer around with them, they could collect information far more easily. This would have little effect on you as a farmer. If however, the farm secretary carried a range of programs which you may want to use, then a bureau-type service could be provided.

Stock firms use computers at present to process their own and farmers' data, but generally for accounting purposes (the comments about accountants would apply). We may soon see changes in this area, however, with stock firms offering bureau type services. Dairy companies are similar to the stock firms.

Data Bank runs a sophisticated bureau service where you can use a computer to carry out relatively sophisticated financial analysis and control. This system uses coded checks indicating categories of income and expenditure as well as journal entries which you make and send in by post or deliver to a bank.

Once you have an understanding of the system and how to interpret the results, it can be most useful. The cost is quite reasonable.

Bureau – Bureau services generally work on a mail-in/mail-out system where you enter the appropriate information on a coding sheet, then send it to the bureau where it is processed. The results are mailed back to you. The type of work and cost varies considerably.

There are several ways you can get access to a computer you can use yourself. These are: full ownership or lease; syndication ownership (group ownership); information system where you use your own terminal; bureau system where you use the computer yourself (basically hire).

Full ownership or lease has the most flexibility. You can do what you want when you want. For a dairy or stud farmer who make almost constant use of the computer, this is the only viable alternative.

Often, a farmer with his own computer may use other computers, such as the bureau service provided by Sheeplan. If real-time data acquisition and control is being considered, your own computer is the only option.

Sharing ownership with others may be possible if you do not need a computer all the time. If you were going to carry out financial analysis and control work, with a small amount of physical recording, then this may be OK.

As in any syndication of equipment, the rules of who can use the computer and when, should be clearly stated before purchase, otherwise disagreements and arguments could ruin good friendships. Always keep the syndication on a business, not a personal basis.

Cheap – This could be a relatively cheap method of getting access to a computer for many farmers. One way to establish a syndicate with minimum cost is to follow the example set by a group of farmers and a programmer in Canterbury. They bought a computer for the local high school (saving the 40% sales tax) and have use of it in the evenings; the school uses it during the day.

Although I know of none operating in New Zealand, information systems could have a place in farming. An information system can take many forms, but in computing, this generally means a large computer which stores a great deal of information on one or more subjects.

It is used like a library. By linking up to the computer with a terminal (keyboard and screen) or another computer, you get access to information in the large computer. The best way to use such a system would be to have your own microcomputer for most of your work, then when you want to tap the information system, you link your computer to the big computer by telephone.

With a bureau system where you use the bureau computer yourself (instead of mailing the information in), you could travel to the bureau office or have a terminal. With computer equipment still relatively expensive, this could be cheaper than using your own computer, although not quite as convenient.

If you are considering any of these alternatives, you must still carefully evaluate the costs and benefits of each system. If you do not own your own computer, the inconvenience of the other alternatives must be valued as a cost.

Next month, we will look at what should be considered when determining the costs and benefits of owning a computer.

Final selection

By JOHN J. VARGO

Last month, we evaluated the vendor proposals and reduced our list to the most promising candidates to supply our small business system. We will now look at the process of making the final vendor selection and implementing the chosen system.

Final selection — Having chosen your short list of potential system suppliers, you now need to determine their ability to deliver what they have promised. Your selection was based on their apparent ability to meet your needs and the cost/effectiveness of their system.

Final evaluation of vendor and system capabilities will include:

- contacting current users for their opinions and experiences with the vendor and the system you are planning to install.
- testing the system with typical input from the proposed user environment.
- evaluating documentation for hardware and software.
- appraising printed reports and CRT displays for suitability of format, quality and flexibility.

A system may appear to meet all your needs and be very effective on paper. But when it is measured in its natural business environment for ease of use, vendor maintenance support etc, it may be found wanting. You need to discover before you buy if the real capabilities match up to the purported ones.

Contacting current users — To obtain names and addresses of current users on your short list, just ask your vendors. They should be pleased to provide such a list. But if they are hesitant, this may be an indication of the quality of service provided in the past.

A phone call may be convenient for initial contact to determine the user's willingness to provide such information. The user's final response however should be in writing so that there are no misunderstandings about your questions, and to ensure clarity of responses.

You may choose to prepare a "user questionnaire" which states the information you require and may provide a scale for response from 1 (very poor) to 5 (excellent). Here are some questions you may want to include:

- How has vendor support been in terms of hardware and software maintenance?
- Has the hardware been as reliable as you expected?
- Has technical and user documentation been up to your expectations?
- Have vendor training sessions been satisfactory and sufficient in scope to prepare your employees for using the new system?
- Have there been any other areas you have found particularly troublesome?
- Any areas you have been particularly pleased about?

Based on the results of the questionnaire, you may want to clarify certain points with the vendor, or include certain items in your contract.

Testing the proposed systems — Testing the proposed systems is the key evaluation tool. A few days of thorough testing at this stage can save months of frustration later. Tests should be performed using samples of actual transactions from your business selected to provide as wide a range of circumstances as possible. Transactions should include very small and very large dollar amounts, as well as erroneous information of every variety.

The testing process will determine the suitability of the system to your working environment, as well as testing its reliability and error detecting capabilities.

Documentation — Since the system you choose will be used for five years or more, it is important the original documentation is sufficient to train future employees as well as provide an ongoing reference for current users. There should be user and technical manuals for hardware and software components.

All user manuals should be clearly written and indexed so that a first-time user can read it and not feel threatened. At the same time, manuals should be complete enough to be a useful reference to the experienced systems designer. Unfortunately, not every manual may meet these guidelines.

Appraisal of output — The printed documents and reports from the system, as well as the screen layouts, should be approved by the manager/users and operator/users. If the people who will be using the system are not satisfied with the final output, we may find the whole system failing. We must always remember the people in the business are a key element in our information system and the success of the installation depends on their enthusiastic support.

Printed reports and screen formats should be easy to follow and as similar to the manual system reports and forms as possible. This requires flexibility in the system and good communication with users. This is also an opportunity to evaluate the quality of the printer proposed for the system, and the quality and stability of the CRT display.

The contract — By this time in the evaluation process, it has probably become apparent which vendor is your top choice. A contract should be drawn up to formalise your choice and spell out the responsibilities of both parties. As a minimum the contract should include:

- a specification of the hardware, software and training to be provided by the vendor
- delivery dates and a program of implementation, and related payment schedule
- a specification of the maintenance program for software and hardware, and the related costs
- any site preparation which might be required for the hardware.

These are minimal suggestions, and it is recommended you contact your solicitor for further advice.

- **Next month, John Vargo will discuss the implementation of a new system.**

The mechanics of a network

By NICK SMYTHE

This article focuses on a few of the major questions and evaluation issues surrounding microcomputer networks. To do this we will refer to several networking systems available for the microcomputer systems that have attracted most interest: the Apply, Poly and BBC Microcomputer. At present the comments are largely secondhand but perhaps at a later date we can return to report on some "hands-on" experience.

1 Is a network cost-effective?

The first part of this issue depends on how many stations you want to have and what sort of disks and printers you want. Basically the more computers the better the economics since for each station you save on a disk and printer. You can use that money either to get better disks or printers or to buy more computers. Nothing is this simple though so go to 3 to see at least one caveat on this comment.

2 Will it lose my flexibility with machines?

Usually no. Most networks enable you to plug machines on or off with little trouble. The machines you take off are as autonomous as the equipment you have. If you have some small spare disk drives then you can take any machine away and use it as an independent micro wherever you want. Later on you can plug it back in. The real question here is whether the extra disk drive is available. An alternative to full detachment is linking a microcomputer in to the network cable at another point. However, there are limits on how far a network cable can stretch without extra equipment.

3 Can I add any number of computers on?

In theory some networks have the capability to recognise a large number of machines. Some of the Apple systems and Econet offer 255 stations for instance. This is a little spurious since as you add machines a nasty phenomenon called system degradation sets in. Essentially this is a micro-electronic traffic snarl-up.

Each user is fine while he is using his own processor but whenever he uses the network to share things he has to compete with other users. Once you get into large numbers of users this competition can slow things down a lot. The number you can sustain depends on your use of the system. If your use is disk or printer intensive the number will be be lower than if most of the work is done by independent machines. One way of looking at a network is that you want users to get a fast response time. Slow response, say to a disk load, may indicate either slow disks or slow transfer along the wires or excessive queues. The last category is the real killer and can occur with surprisingly few users in some cases.

BITS . . & . . . BYTES . .
NEW ZEALAND'S PERSONAL COMPUTER MAGAZINE

The BITS & BYTES Computer Book Club

You've got a new ZX81? Then here's the book to go with it!

Douglas Hergert's Your Timex Sinclair 1000 and ZX81 starts with the real fundamentals, such as how to connect the machine to the TV set and a cassette recorder. It takes you through home to write programs for graphics, calculations, games, and more. It introduces you to BASIC.

The book also has ready-to-run programs including how to turn your ZX81 into a super calculator; how to make bar graphs to help you calculate home finances, and how to draw pictures on your TV screen.

This is really value buying at our cut price of $11.85. This is 65c less than normal retail, and you are credited with a bonus point which can be used to buy the great bonus-point-only specials coming up later in the year.

Get your order in now. Use the coupon on the next page.

Just look at these prices! You get a special credit in addition! And they come right to your home by mail (We don't even charge for postage or packing).

Specials for bonus-points holders begin in the May edition of "Bits & Bytes"

How it works —

Once you've bought a book, you're in the club. So just pick out the books you want, fill in the coupon on the next page and post it in.

What you get —

We are offering savings on the cash you pay for each book PLUS we give credit bonus points on each purchase. These are stored to your credit, and a number of times a year we will offer special great buys for cash plus these points. These bargains will be available only to those with credit points. You get the books post free!

This month's offer is only a beginning. We plan to widen the range of titles until we can offer one of the world's best ranges of computer books.

Computer books to read and use

Don't, or How to Care for your Computer Rodney Zaks
An easy, entertaining guide to computer and peripheral preservation. Specific advice for the computer, floppy discs, hard discs, the CRT terminal, the printer, tape units, the computer room, software, and documentation. In the words of "Popular Computing" this book is "... cheap insurance".

Sybex — Our price $22.75 **Save $1.20**
and earn two bonus points

From Chips to Systems: An Introduction to Microprocessors Rodney Zaks
A superb, fast-paced journey through the history of microprocessors, the microprocessor chip itself, its support components, and the design of an actual microcomputer system. You'll find out that microprocessors were developed by accident rather than design, that early structural errors have become today's features and, most importantly, how easy it is to understand microcomputers. This is what "Elementary Electronics" said about this book: "One of the best introductory texts to the hardware end of things."

Sybex — Our price $30.35 **Save $1.60**
and earn three bonus points

Applesoft BASIC A Teach-Yourself Introduction Barrie M. Peake
A manual for New Zealanders. A one-book method of learning BASIC with the Apple, instead of picking information from two or three. "You get quite a wide view of the Apple system with one book instead of three or four," writer/reviewer Mike Wall in the March issue of "Bits & Bytes". Model answers. Inquiries regarding terms for class sets welcomed.

John McIndoe — Our price $9.45 **Save 50c**
and earn a bonus point

Mastering Computer Programming P.E. Gosling
A complete, self contained course from Britain for self study at home, or for use in the classroom. In one book, the essential information to learn programming. The language taught is BASIC. The author, a former lecturer in computing, runs a computer services company.

Macmillan Master Series — Our price $9.45 **Save 50c**
Plus 1 bonus point

Get the order in today!

Pick out your books, fill in the coupon, and get it in the mail today. Stocks are not unlimited, and the last in may have to wait.

FREE POST TO YOUR HOME, PLUS -BONUS POINTS!

GET THIS IN THE MAIL TODAY:
To buy, cut out this coupon and post it to:

Freepost 125
"Bits & Bytes" Books
Box 827
Christchurch

(If you address it in this way no stamp is needed)
Pay with order: Cheque, money order or fill in the Bankcard/Visacard boxes.

Payment:
SIGNATURE: .....
NAME: .....
ADDRESS: .....
(zone) .....

I enclose payment for the above order.

☐ Visacard
☐ Bankcard

Card holder signature .....
Date card expires .....
Cheque ☐ Cheque No. .....
Postal order ☐ Postal note ☐

Please note any areas of special interest: .....

Machine(s) in which you are especially interested .....

Please tick the appropriate boxes: Please send me, post free, the following books:	Our Price	You Save	Bonus pts
☐ Your Timex Sinclair 1000 and ZX81 (Hergert)	11.85	.65	1
☐ Microcomputer Design and Troubleshooting (Zumchak)	30.75	3.35	3
☐ Interface Projects for the TRS 80 (III) (Hallgren)	23.35	1.25	2
☐ Quick Keyboarding (Alexander)	6.50	.45	1
☐ Introduction to T-Bug (Inman)	15.15	.80	2
☐ From Chips to Systems (Zaks)	30.35	1.60	3
☐ Inventory Management for Small Comp. (Atkinson)	24.65	1.35	2
☐ Introduction to TRS-80 Graphics (Inman)	18.95	1.10	2
☐ Basic BASIC-English Dictionary (Noonan)	20.80	1.15	2
☐ The Tenderfoot's Guide to Word Processing (Chirlian)	18.95	1.00	2
☐ Inside Basic Games (Materosian)	25.18	1.32	2
☐ Don't, or How to Care for you Computer (Zaks)	22.75	1.20	2
☐ Applesoft BASIC (Peake)	9.45	.50	1
☐ Mastering Computer Programming (Gosling)	9.45	.50	1
☐ Mastering Computers (Wright)	9.45	.50	1
☐ Beginning BASIC (Gosling)	11.35	.60	1
☐ BASIC programming on the BBC (Cryer)	19.25	1.05	2
☐ Start with BASIC on the VIC 20 (Monro)	19.25	1.05	2
☐ Learning LOGO on the Apple II (McDougall)	18.05	1.00	2
☐ Atari Games and Recreations (Kohl et al)	27.00	1.45	2
☐ Basic Computing: A Complete Course (Crawford)	24.65	1.31	2
☐ WordStar made easy (Ettlin)	23.85	1.25	2
☐ VisiCalc: Home and Office Companion (Castlewitz)	31.95	1.65	3
☐ Discovering FORTH (Hogan)	29.60	1.60	3
Total payment enclosed			

Put them on a shelf by the keyboard

Mastering Computers G.G.L. Wright
A good introduction for the interested, or for use as a text in a computer-appreciation course. Covers the field from the basis of mainframes, through such everyday applications as bar coding to the possibilities of videotex and an electronic money system. Well written in a concise, British style, and well illustrated.

Macmillan Master Series Our price $9.45 **Save 50c**
Plus 1 bonus point

Microcomputer Design and Troubleshooting
Eugene M. Zumchak
Considers every aspect of microcomputer design from the idea to the working system. Controller functions, the development system, read/write timing, good hardware design, the computer system, hardware testing and troubleshooting and the three basics of software design: documentation, philosophy, and technique.

Howard Sams Our price $30.75 **You save $3.35**
and earn three credit points

Basic Programming on the BBC Neil and Pat Cryer
You've seen the machine on television, and this is the book prepared to go with the programme. It's designed for the new BBC, and teaches how to write programs, draw and animate pictures and graphics in full colour, design sound effects and program games. Detailed glossary.

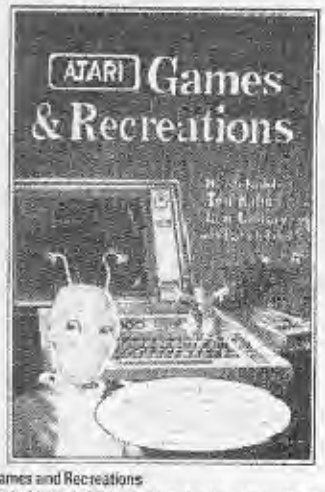
Our price $19.25 **You save $1.05**
and earn two bonus points

Interface Projects for the TRS-80 (Mod III)
Richard C. Hallgren
For the user with some computer experience. A series of easily built interface projects allowing the discovery of the computer's capabilities. Many fully tested practice hardware projects including review of data transfer formats, analog-to-digital conversion biofeedback projects, and controlling a video playback device.

Spectrum Our price $23.25 You save $1.25
and earn two bonus points

Quick Keyboarding Vonnie Alexander
Sub-titled "Competent Keyboarding in 6 Hours", this book by New Zealander Vonnie Alexander has a unique method for teach-yourself competent keyboarding. A wall chart of finger positions is included.

Methuen Our price $6.50 You save 45c
and earn one bonus point



Atari Games and Recreations
By Herb Kohl, Ted Kahn, and Len Lindsay. Provides hours of pre-programmed games to play. But it also gives instruction readers need to improve on these games and create more complex and challenging games of their own. How to tell fortunes, compose songs, guess riddles, do word puzzles, and animate cartoon characters on the Atari.

Reston Our price $27.00 You save $1.45
and earn 2 bonus points

Books help you get the most out of your computer. Be in the club!

Buy now! These are current prices only

BITS..&...
..BYTES..

Introduction to TRS-80 Graphics Don Inman

One for all the 80 users! A complete introduction to the basics of graphics programming on the TRS-80, using dozens of examples. It begins with the most basic concepts of line drawing and leads to geometric shapes, moving figure animation, and other more advanced topics.

dilithium Press — Our price $18.95. Save $1.10 and earn two bonus points

The Tenderfoot's Guide to Word Processing Barbara Chirlain

Can I use a word processor in my profession or small business? What do I need in the way of equipment? Will a word processor be of use to me? These are just a few of the questions this book will answer.

dilithium Press — Our price $18.95. Save $1 and earn top bonus points

Your Timex Sinclair 1000 and ZX81 Douglas Hergert

For Sinclair users: Takes you from the very beginning and explains in simple, everyday language how to use your ZX81 to its fullest capabilities. How to connect your TV and cassette recorder and make them work together. Giving commands, writing your own programs for graphics, calculations, games, and more. Ready-to-run programs include a supercalculator; bar graphs for home budgeting; drawing pictures on the TV screen. A good book for the new user.

Sybex — Our price $11.85. You save 65c and earn a bonus point

Inside Basic Games Richard Mateosian

A fun introduction for anyone beginning programming. The author has chosen the medium of games to teach readers how to design error-free, interactive BASIC programs. Computer games are described in detail, then explained and analysed to illustrate how the games were developed, Rules, algorithms, and coding differences from the TRS-80, Apple II, and PET are also included.

Sybex — Our price $25.18 Save $1.32 and earn two bonus points

Basic Computing: A Complete Course Tim Crawford

From McGraw Hill (Canada). Provides a broad-based introduction to computer science and data processing, suitable for a variety of levels in high schools, universities or industry. Begins with questions such as What is a Computer? Where Did Computers Come From? Gives detailed coverage of computer languages, programming, program structure, logic, testing, documentation and program maintenance.

McGraw-Hill — Our price $ 24.65 Save $1.31 and earn two bonus points

WordStar Made Easy Walter Ellin

In 14 easy lessons, this handbook demonstrates all the powerful features of the MicroPro WordStar system. Lessons cover everything from loading, using the unique printing, text manipulation, and formatting functions. Spiral bound, it is ideal to use while working at the terminal.

Osborne — Our price $23.85 You save $1.25 and earn two bonus points

VisiCalc: Home and Office Companion David Castelwitz, et al

Provides extensive coverage of the popular VisiCalc program, and can be used by the novice and expert alike. For the novice, it offers 40 models that can be used immediately for personal and business applications. For the expert it is a source of new ideas and techniques.

Models created with Apple 3.2 version, but will perform just as well on machines with other versions.

Our price $31.98 Save $1.65 and earn those bonus points

Discover FORTH: Learning and Programming the FORTH Language Thom Hogan

Whether you are a beginner seeking information on this multi-faceted programming language or a serious programmer already using FORTH this book is a reference that should not be overlooked. Describes FORTH syntax, specifically applicable to both FORTH 79 and FIGFORTH.

Our price $29.90 You save $1.60 and earn three bonus points.

Learning LOGO in the Apple II McDougall, et al

LOGO is a Piaget-based way into computing. A multi-purpose language. Non-technical, learning by doing.

Prentice-Hall (Australia) — Our price $18.05 You save $1.00 and earn two bonus points.

POST YOUR SUBSCRIPTION TODAY!!

NAME: .....

ADDRESS: .....

.....

..... ZONE: .....

I enclose payment for one year's subscription to BITS & BYTES

Adult $8 ☐ School $6 ☐ School name .....

Form .....

PAYMENT:

Visacard
Bankcard

Card Holder Signature .....

Date Card Expires .....

Cheque ☐ Cheque No. .....

Postal order ☐ Postal note ☐

7

Please answer the following questions:

My interest is:

Farming
Education
Competitions
Business
Professional
Hobby use
General news
Programs
Beginner

My field is:

Professional (medical, accounting etc)
Business
Education (lecturer, teacher etc)
Consumer (hobbyist etc)
Farming
Student
Other

Do you use a microcomputer Yes/No

If yes, which brand .....

BACK COPIES

Please send me the following back issues at $1 each plus 50c postage and packaging each.

Issue No(s): - - - - - - - - - - - -

Number of Copies: - - - - - - - - -

(Please send payment with order. Note; no more copies of Issue No. 1 or 2 are available and other stocks are limited)

No stamp is required if you write this address on your envelope

Freepost No. 125
BITS & BYTES
P.O. Box 827
Christchurch.

4 Will I need expensive peripherals?

This again depends on your requirements. Networks become unattractive when they slow right down. Although the worst culprit is user competition, clearly better peripherals can ease this a lot. Hard disks, for instance, can perform at an average of 10 times the speed of floppy disks and that can clear a lot of queued jobs quite speedily.

The disk drive is particularly important if you solve your printer demand problems by saving printing requests temporarily to disk. The currently cheaper 5¼in floopy drives are slow and have a small capacity. The spooling can run out of disk storage room. Most common 5in drives are in the 100-180 kilobytes range. Recent ones get to 400K. The 8in floppy is faster and larger. Best of all is the hard disk which has access times typically an order lower than any floppy and stores from 5 megabytes (million bytes) upward. Hard disks seem made for networks. Furthermore cost competition overseas is driving the price down very fast. A hard disk with 5 megabytes should be available for under $5000 (educational price) in 1983. In fact the Fanasonic 3001 16-bit computer now marketed here is already rumoured to be more than shaving that threshold.

While we are on hard disks it is worth noting that the sort schools can afford do not have changeable platters as a floppy drive does. The disk is sealed in its container. The corollary of this is that you need a back-up system tc hold your files in case the hard disk crashes. This may be a tape or a floppy drive or both. Formac Marketing in Palmerston North is working on an interface to permit back-up storage to a videotape recorder and this may be a cheap possibility. The Corvus system for the Apple has a similar facility. Running a hard disk means more thought is given to a lot of matters regarding file security and who has what file space. This is such that most networks would want a floppy drive and a hard disk drive just for user flexibility.

5 What is a Megabaud network?

Another reason that networks are slow is that their transfer rate is slow. The cable in a network allows the transfer of data from various devices attached to it. The speed of this transfer is governed by the cable and by the transmission and reception ports in the computer and peripheral. Some networks are slow and some very fast. Some will meet small needs and some major ones. You can get an idea of what is involved by considering the baud rate of networks. A baud is the transfer of one bit (an eighth of an 8-bit byte) a second. Network transfer speeds in our range vary between 9600 baud and 1,000,000 baud with several on 250,000. This translates to 1200, 125,000 and 30,000 bytes a second or, in practical terms, the transfer of an 8K file (say a Poly or Apple screen image) would take 6+ seconds, 1/15 second, or about 1/4 of a second. As we noted there are other overheads in timing. These may be so significant as to dwarf transfer times. A small floppy loading the 8K to the network would take so long loading on any network that the

Continued page 32

```
0 POKE 2,0:POKE 3,6:POKE 9,2:POKE 1536,7
6:POKE 1537,64:POKE 1538,185:TRAP 17000
1 GRAPHICS 0:POKE 752,1:SETCOLOR 2,6,1:P
OSITION 0,10:? ,"    ESCAPE MAZE":POSITI
ON 0,12:? ,"   G.C. ROBERTS"
2 POSITION 0,14:? ,"         1982"
3 FOR I=1 TO 255 STEP 4:SOUND 0,I,8,10:G
OSUB 11000:NEXT I
5 FOR T=1 TO 10:SETCOLOR 2,6,1:FOR D=1 T
O 50:NEXT D:SETCOLOR 2,6,7:FOR D=1 TO 50
:NEXT D:NEXT T:SOUND 0,0,0,0
6 GRAPHICS 0:POKE 752,1:SETCOLOR 2,4,6:S
ETCOLOR 1,0,0:RESTORE
7 TIME=0:FOR T=1 TO 23:? "EASY! NOT AS E
ASY AS IT LOOKS!"
8 FOR R=1 TO 5:SOUND 0,INT(RND(1)*256),1
0,10:NEXT R:NEXT T:SOUND 0,0,0,0
9 REM *   <<< MAZE 1 >>>
10 GRAPHICS 7+16:COLOR 2:GOSUB 2000
12 DATA 0,0,159,0,159,0,159,95,159,95,0,
95,0,95,0,50,0,40,0,0
13 DATA 10,10,30,10,40,10,80,10,90,10,15
0,10,2,20,10,20,50,20,90,20,100,20,150,2
0,30,30,40,30,60,30,70,30
14 DATA 80,30,90,30,120,30,150,30,40,40,
50,40,70,40,100,40,110,40,150,40,30,50,5
0,50,90,50,100,50,110,50,150,50
15 DATA 40,60,60,60,70,60,80,60,100,60,1
50,60,4,70,30,70,54,70,90,70,100,70,150,
70,20,80,40,80,60,80,76,80
16 DATA 90,80,150,80,24,90,50,90,70,90,8
0,90,10,20,10,80,20,10,20,60,20,80,20,90
,30,20,30,40,30,50,30,60
17 DATA 40,10,40,30,40,60,40,80,50,20,50
,50,50,64,50,94,60,30,60,70,60,80,60,95,
70,2,70,10,70,40,70,50
18 DATA 80,30,80,60,80,70,80,90,90,10,90
,30,90,50,90,70,90,74,90,95,100,30,100,4
0,110,30,110,40,999,999
27 X=156:Y=48
28 COLOR 1:SETCOLOR 2,0,0
29 GOSUB 9000
62 IF MAZE=1 THEN IF X<60 AND Y<40 THEN
GOTO 27
63 IF X<1 AND Y<50 THEN GOTO 3000
64 GOSUB 7000
65 COLOR 1:SOUND 0,0,0,0:PLOT X,Y:COLOR
3:PLOT J,K:GOTO 28
1000 GOSUB 16000
1010 SOUND 0,64,10,8:GOTO 27
2000 READ X,Y:IF X<>999 THEN PLOT X,Y:RE
AD X,Y:DRAWTO X,Y:GOTO 2000
2010 RETURN
```

A-Mazing game for Atari users

A game for Atari users called Escape Maze by G.C. Roberts of Te Kuiti.

Requirements: Atari 400/800; 16K memory; Basic Cartridge; one joystick.

Game Object: To escape through all three mazes.

Difficulty: Intermediate.

Language: Basic.

Graphics & Sound: Yes.

Programme Line Analysis:—

Lines 1-8: Introduction. Add the following lines if you want SYSTEM RESET to restart the game.

0 POKE 2,0:POKE 3,6:POKE 9,2:POKE 1536,76:POKE 1537,64:POKE 1538,185:TRAP 17000

17000 RUN

Lines 9-65: Commands for first maze. Gosub 2000 reads Data lines (12-18) to draw maze. Gosub 9000 (-9090) gives joystick commands which control the moving pixel. These commands are same for mazes 2 & 3 also. Line 9000 specifically gives time limit for all 3 mazes. Line 62 tests x,y co-ordinates for the invisible walls which if positive sends pixel back to start. Line 63 tests to see if pixel is at door of first maze. Line 64 with Gosub 7000 tests for pixel hitting wall (if positive Gosubs 1000-1010); Gosub 16000 records number of wall hits. Line 65 draws pixel with colour 1 while followed by colour 3 so as not to leave a trail.

Lines 3000-4020: Graphics & sound commands for escaping first maze.

Lines 4090-5180: Commands for second maze. Line 5157 is a test for sending pixel through to the third maze. Line 5159 with Gosub 14000 tests for pixel hitting wall. If positive Gosub 13000-13010 gives the graphics & sound routine. If negative lines 14020-14030 randomly draw a wall (14030) in maze at a low probability rate (14020). Gosub 15000 randomly places pixel somewhere else in maze as a result of being blasted & thrown from wall.

Lines 5190-6090: Commands for third maze. Line 6075 tests to see if pixel is at door of third maze. If positive lines 8000 to 8070 give graphics & sound commands for conclusion & replay option. Line 6080 with Gosub 7050 tests for pixel hitting wall. If positive sends player back to start of first maze.

```
3000 X=1:Y=45:FOR T=1 TO 3:FOR C=1 TO 14
:SOUND 0,64,C,C:SETCOLOR 1,C,C:FOR D=1 T
O 35:NEXT D:NEXT C
3005 FOR W=1 TO 20:NEXT W:NEXT T
3010 FOR C=0 TO 15 STEP 3:FOR D=1 TO 10:
NEXT D:SETCOLOR 4,C,7
3020 FOR P=243 TO 31 STEP -7:FOR D=1 TO
5:NEXT D:SOUND 0,P,10,9:NEXT P:NEXT C
3900 GRAPHICS 0:POKE 752,1:SETCOLOR 2,6,
4
4000 POSITION 0,10:PRINT ,,"CONGRATULATI
ONS!"
4005 PRINT ,"YOU FOUND A WAY AROUND THE
 INVISIBLE WALLS!"
4020 SOUND 0,0,0,0:FOR T=1 TO 900:NEXT T
4090 REM *   <<< MAZE 2 >>>
5000 HIT=0:MAZE=0:TIME=0:GRAPHICS 7+16:S
ETCOLOR 1,8,10:COLOR 2:GOSUB 2000
5100 DATA 0,0,159,0,159,0,159,95,159,95,
0,95,0,95,0,50,0,40,0,0,20,10,50,10,60,1
0,150,10,1,20,30,20,40,20,66,20
5105 DATA 80,20,140,20,20,10,20,20,30,0,
30,10,10,30,30,30,90,30,100,30,44,80,50,
80
5110 DATA 30,30,60,30,110,30,140,30,10,4
0,50,40,110,40,150,40,20,50,56,50,80,50,
90,50,110,50,150,50,0,60,20,60
5120 DATA 24,60,60,60,100,60,140,60,20,7
0,40,70,50,70,76,70,80,70,140,70,10,80,4
0,80,60,80,150,80
5130 DATA 10,40,10,56,20,50,20,95,30,20,
30,30,50,30,50,40,50,64,50,95,60,10,60,6
0,70,20,70,70,80,20,80,50
5140 DATA 80,60,80,70,90,34,90,66,100,30
,100,60,110,30,110,40,140,20,140,30,140,
60,140,70,150,10,150,40
5150 DATA 40,14,40,20,60,84,60,95,70,0,7
0,10,110,50,110,56,150,50,150,80,999,999
5155 X=156:Y=48
5156 COLOR 1:SETCOLOR 2,0,0
5157 IF X<50 AND Y<50 THEN GOTO 6000
5158 GOSUB 9000
5159 GOSUB 14000
5180 COLOR 1:SOUND 0,0,0,0:PLOT X,Y:COLO
R 3:PLOT J,K:GOTO 5156
5190 REM *   <<< MAZE 3 >>>
6000 TIME=0:GRAPHICS 7+16:SETCOLOR 1,4,1
0:COLOR 2:GOSUB 2000
6005 DATA 0,0,159,0,159,0,159,95,159,95,
0,95,0,95,0,60,0,50,0,0
6010 DATA 10,10,50,10,60,10,90,10,100,10
,140,10,10,20,60,20,80,20,130,20,140,20,
150,20,20,30,86,30,90,30,110,30
6020 DATA 130,30,140,30,1,40,20,40,50,40
,140,40,20,50,30,50,54,50,130,50,10,60,2
0,60,50,60,96,60,100,60,110,60
6030 DATA 54,70,90,70,120,70,136,70,20,8
0,30,80,40,80,60,80,70,80,140,80,10,90,2
0,90,34,90,66,90,70,90,150,90
6040 DATA 10,10,10,30,10,50,10,86,20,40,
20,60,20,64,20,90,30,34,30,76,30,80,30,9
5,40,30,40,80,50,20,50,30
6050 DATA 50,40,50,76,50,80,50,90,60,1,6
0,10,70,14,70,30,70,64,70,95,80,10,80,20
,90,30,90,40,10,30,10,40
6060 DATA 100,14,100,30,100,44,100,70,11
0,60,110,80,120,20,120,36,120,60,120,70,
140,10,140,30,140,40,140,70
6070 DATA 130,50,130,60,150,20,150,50,15
0,60,150,90,999,999
6072 X=156:Y=55
6074 COLOR 1:SETCOLOR 2,0,0
6075 IF X<3 AND Y<59 THEN GOTO 8000
6076 GOSUB 9000
6080 GOSUB 7050
6090 COLOR 1:SOUND 0,0,0,0:PLOT X,Y:COLO
R 3:PLOT J,K:GOTO 6074
7000 LOCATE X,Y,A:IF A=2 THEN GOTO 1000
7010 RETURN
7050 LOCATE X,Y,A:IF A=2 THEN RESTORE :G
OTO 6
7055 RETURN
8000 X=1:Y=55:FOR S=1 TO 2:FOR I=255 TO
1 STEP -4:SOUND 0,I,10,10
8005 GOSUB 10000:NEXT I
8010 SETCOLOR 2,INT(RND(1)*16),0:NEXT S:
FOR I=255 TO 1 STEP -4:SOUND 0,I,8,10:GO
SUB 11000:NEXT I
8015 WAIT=10:FOR T=1 TO 20:SOUND 0,INT(R
ND(1)*256),10,10:GOSUB 12000:NEXT T:SOUN
D 0,0,0,0
8020 WAIT=3:FOR I=1 TO 255 STEP 4:SOUND
0,I,8,10:GOSUB 12000:NEXT I
8025 SOUND 0,0,0,0
8027 GRAPHICS 2:SETCOLOR 4,4,6
8028 POKE 709,12
8029 POKE 710,0
8030 ? "PRESS THE TRIGGER TO PLAY AGAIN
"
8040 FOR Y=1 TO 500:IF STRIG(0)=0 THEN R
ESTORE :GOTO 6
8050 NEXT Y:GRAPHICS 0:POKE 752,1
8060 POKE 709,12:POKE 710,0:POSITION 0,1
2:? ,"       THE END"
8070 GOTO 8070
9000 TIME=TIME+1:IF TIME=600 THEN GOTO 6
9010 J=X:K=Y:IF STICK(0)=14 THEN Y=Y-1
9020 IF STICK(0)=6 THEN X=X+1:Y=Y-1
9030 IF STICK(0)=13 THEN Y=Y+1
9040 IF STICK(0)=5 THEN Y=Y+1:X=X+1
9050 IF STICK(0)=9 THEN Y=Y+1:X=X-1
9060 IF STICK(0)=11 THEN X=X-1
9070 IF STICK(0)=10 THEN Y=Y-1:X=X-1
9080 IF STICK(0)=7 THEN X=X+1
9090 RETURN
10000 FOR W=1 TO 5:NEXT W:RETURN
11000 FOR W=1 TO 5:NEXT W:RETURN
12000 FOR W=1 TO WAIT:NEXT W:RETURN
13000 GOSUB 15000
13005 FOR I=1 TO 255 STEP 20:SOUND 0,I,8
,10
13010 FOR T=1 TO 15:POKE 710,32+T:NEXT T
:NEXT I:SOUND 0,0,0,0
13012 GOSUB 16000
13015 RETURN
14000 LOCATE X,Y,A:IF A=2 THEN GOSUB 130
00
14020 IF RND(0)*200>1 THEN RETURN
14030 TX=X:TY=Y:COLOR 2:GOSUB 15000:PLOT
 X,Y:GOSUB 15000:DRAWTO X,Y:X=TX:Y=TY:RE
TURN
15000 X=INT(RND(0)*159):Y=INT(RND(0)*95)
:RETURN
16000 HIT=HIT+1:IF HIT=4 THEN GOTO 6
16010 RETURN
17000 RUN
```

A graphic game

By BRIAN BULLEN

When I first got my VIC, I started at the front of the guide and worked as fast as possible through it. Every so often I'd come across something that would make me think "must come back to that".

One of these was the section on animating with Peeks and Pokes (p61-65). It reminded me of some of the first television games which appeared in New Zealand.

I want to show how their program can be turned into a simple game, and introduce a few features of the VIC we haven't yet looked at.

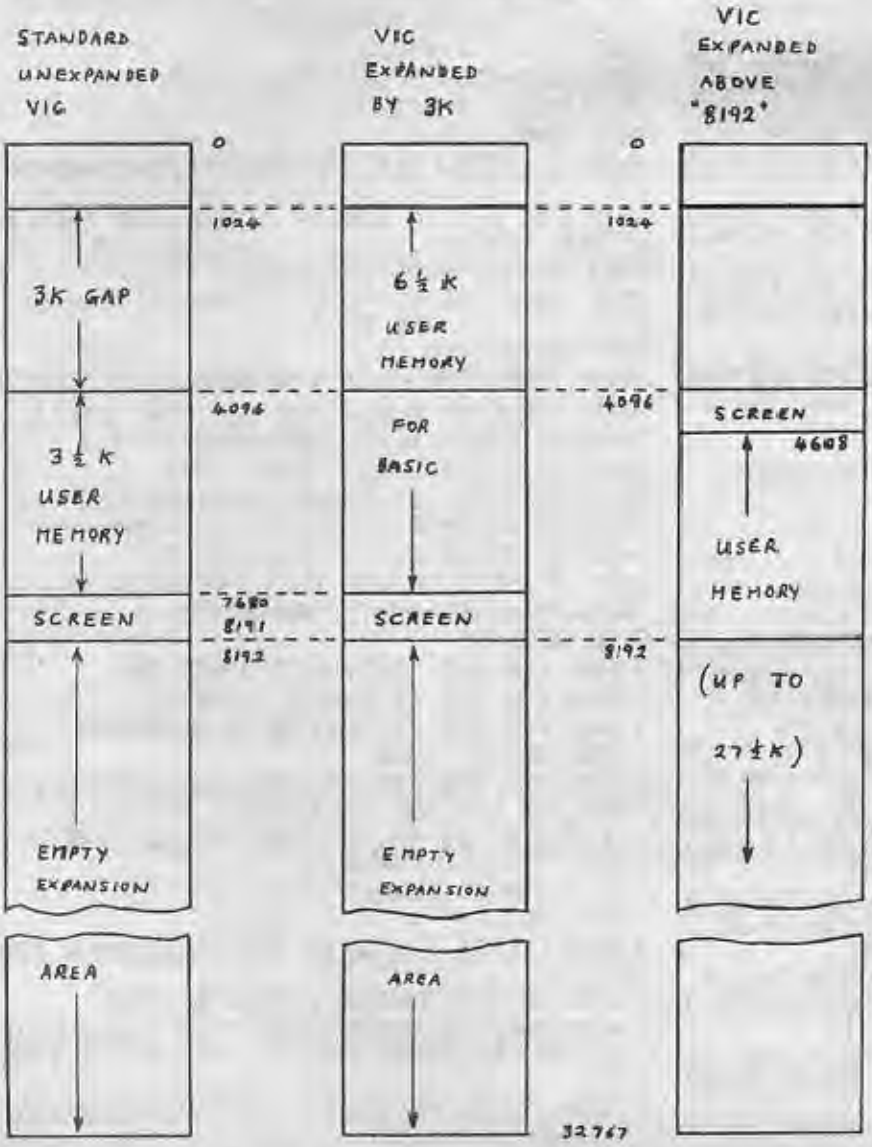


FIG. 1 Memory map of bottom 32K of three versions of VIC Memory Expansion.

I suggest you reread the section in the guide. Be warned there is a mistake on p62, line 6 - the words row and column are interchanged.

Now for the game. To their program, we will add a bat with which to hit the ball, controlled by the f1 and f7 keys, a record of how many times the ball gets past us, and a timer to see how long we can survive. Enter the program listed below and run it.

As it stands at the moment, it's a rather hard and unsatifactory game. To get the bat to move fast enough, you have to peck away at the keys like a woodpecker, but we'll do something about that later on. First let's have a look at some of the different parts of the program.

Interest

The first bit of interest is the TI$="000000" in line 20. This is setting the VIC's built-in clock to zero so that we can time how long we play the game.

Lines 230 and 240 then use TI$ and extract the minutes elapsed using MID$(TI$,3,2), which picks out two characters of TI$ starting at the third character in. Then the seconds are extracted using RIGHT$(TI$,2) which picks out the two rightmost characters of TI$.

This sort of technique can be used whenever you want to time something. You could even use it to turn your VIC into the most expensive digital clock on the market.

The next lines of interest are 140-160. Line 140 uses the GET statement to see which, if any, key has been pressed. In most applications, the GET statement is used in a loop which makes the program wait until a key has been pressed. 140 GET P$:IF P$ = ""THEN 140 is the normal form for this.

In our case, we most definitely don't want the program to wait. Try changing line 140 to that above and you will see what I mean. Line 150 checks to see if the f1 key has been pressed and, if so, moves the bat up the screen. Line 160 does likewise for the f7 key and moves the bat down, PY being the variable which determines the bat position.

You could use any keys you wished. You would just have to change the 133 and 136 CHR$

values. For instance, if you wanted to use U for up and D for down, the values would be 85 and 68 respectively. (Appendix J in your guide has the values for all the keys). Alternatively you could use a statement such as IF P$ = "U" THEN... This has the same effect.

Checks

Line 170 checks to see if the bat has hit the ball (97 is the CHR$ code for the bat). If it has hit, it reverses the ball's direction, makes the appropriate noise and goes back to line 60 to work out the ball's new position without putting the ball on the screen.

Now to deal with that "woodpecker" problem. The easiest way is to change the two control keys to N and Y. Then change lines 150 and 160 to start IF PEEK (37153) = 239 THEN... and IF PEEK (37153) = 247. Now you don't have to peek at the keys but simply hold down the desired key.

This effect occurs because we are taking advantage of the VIC's polled keyboard (that's another article in itself).

Make these changes and run the program. The game is far more playable, isn't it?

However, there is still one problem to sort out. You should have noticed that when you finished the game, it came up with the normal question, "ANOTHER GAME" and then a whole lot of Ns and Ys.

Key

The key to what is going on here lies in the fact there are exactly 10 Ns and Ys. This is because the VIC has a keyboard buffer which stores the keys as they are pressed and

Continued page 38

Program listing

```
10 PRINT"(CLEAR)"
20 POKE 36879,105:POKE36878,15:TI$
   ="000000"
30 X=1:Y=10:DX=1:DY=1:PY=10
40 POKE (7680+X+22*Y),81
50 FORT=1TO10:NEXT
60 POKE(7680+X+22*Y),32:POKE(7680
   +21+22*PY),32
70 X=X+DX
80 IFX=0THENDX=-DX:POKE36876,220
90 IFX>21THENGOSUB200
100 Y=Y+DY
110 IFY=0ORY=23THENDY=-DY:POKE
   36876,230
130 POKE36876,0
140 GETP$
150 IFP$=CHR$(133) THENPY=PY-1:
   IFPY<0THENPY=0
160 IFP$=CHR$(135) THENPY=PY+1:
   IFPY>22THENPY=22
165 POKE (7680+21+22*PY),97
170 IFPEEK(7680+X+22*Y)=97THENDX
   =-DX:DY=-DY:POKE36876,180:
   GOTO60
180 GOTO40
200 C=C+1
210 Y=INT(RND(1)*23):X=1:IFC=20
   THENPY=10:RETURN
220 POKE36879,27:PRINT"(CLEAR,
   DOWN,3RIGHT)GAME OVER"
230 PRINT" YOU SURVIVED FOR ";
   MID$(TI$,3,2)
240 PRINT"MINUTES & "; RIGHT$ (TI$,
   2);" SECONDS"
250 INPUT"ANOTHER GAME";R$
260 IFLEFT$ (R$,1)="Y" THENC=0:
   GOTO10
270 PRINT"GOODBYE FOR NOW":END
READY
```

NEW

Panasonic

FUNCTIONAL SPECIFICATIONS

- Microprocessor
  Model: MN1800 (equivalent to 6802)
  Clock frequency: 890KHz
  System Reset Function
- Memory
  ROM: 8K Bytes
  RAM: 16K Bytes
  Video RAM: 1K Bytes

- Keyboard
  System: Software scanning
  Keys: 5-shift key mode with 45 keys, SHIFT key and CTRL key
- Display Interface
  Screen size: 24 lines × 32 characters
  Characters: 64 characters with 6 × 7 dot matrix
  64 semi-graphic characters with 8 × 8 dot matrix
  Characters & symbols specified by user: 32 characters with 8 × 8 dot matrix
  Attribute: Inverted display function
  Composite video signal: with 75 ohms, 1V p-p or with RF flip-flop converter
- Cassette Interface
  System: FSK system 1,200Hz (space), 2,400Hz (mark)
  Baud rate: 600 Bauds
- AC Adaptor
  Input Voltage: AC 110V, 120V or 220V ± 10%, 50/60Hz
  Output Voltage: DC 17V, 7.8V and −8V
  Power Consumption: 12.5W
- RF Modulator

Panasonic
just slightly ahead of our time

THE MICROCOMPUTER ELECTRONIC COMPANY LTD
IN ASSOCIATION WITH FISHER & PAYKEL LTD

27 GREAT SOUTH ROAD, NEWMARKET,
P.O. BOX 9224, AUCKLAND 1, NEW ZEALAND
TELEPHONE (9) 504-774.

Living with accidents

By GORDON FINDLAY

In my job, and because computing has become my main interest outside work as well, I see a lot of software. Much of it is pretty hard to work with.

This includes a lot of commercial software — it is hard to love a system in which an accidental press of a particular key can destroy all of the last two hours' work. Yet this is just what happens with Apple Logo — hit RESET (which is right next door to RETURN, for heaven's sake) and you are completely sunk.

And what about the programs which die when you make a little typing error?

A little programming can get around these problems.

Here are some techniques for making your programs easier to work with. The main thing is to make input as painless as possible. This usually means avoiding the INPUT statement, which is far too inflexible for most purposes.

This first routine is pretty obvious — it simply scans the keyboard until a key is pressed, then returns to where it was called, with ZX$ containing the key pressed:

```
10010 ZX$=INKEY$: IF ZX$="
  "THEN 10010 ELSE RETURN
```

Notice there is nothing at all between the two quote marks — we are comparing ZX$ to the null string, not a blank.

Input given by the user of a program should always be checked. A program will usually crash if a string is given rather than a number. Using the INPUT statement isn't all that helpful — here is a subroutine you can use to input one key, which must be a digit. If a letter or other character is hit, it is ignored. The theory here is that most typing errors occur by hitting keys near the correct one; and letters are often typed accidently in numerical input.

```
10020 ZX$=INKEY$: IF (ZX$ «
  "0") OR (ZX$» "9") THEN
  10020 ELSE 2X=VAL (ZX$):
  RETURN
```

In this case, the digit is returned as the variable ZX, and as the string ZX$.

Now we can elaborate this idea into the following subroutine, which inputs a number which it returns as ZZ. This can be used as a subroutine: instead of using INPUT AS, use GOSUB 10030: AS = ZZ.

In this subroutine, only digits, the decimal point, and the ENTER (or NEW LINE) key are acted on — everything else is regarded as a typing error, and ignored.

```
10030 CR$=CHR$(13)
10040 ZZ$=""
10050 ZX$=INKEY$:IF ZX$="
  THEN 10050 ELSE IF ZX$=CR$
  THEN ZZ=VAL (ZZ$): RETURN
10060 IF (ZX$»"9") OR (ZX$ «
  ".") OR (ZX$="/")THEN
  10050 ELSE ZZ$=ZZ$+ZX$:
  PRINTZX$;:GOTO 10050.
```

This subroutine works by filtering out all the unwanted characters, and jumping the rest together into a string. The VAL function of Level II BASIC functions quite happily for strings, including decimal points — a point (pardon the pun) which not all the books seem to be clear on.

Of course, this only accommodates positive input. The next subroutine accepts a negative sign as first character, allowing negative values input. Again, the subroutine returns ZZ:

```
10030 CR$=CHR$(13)
10040 ZZ$=""
10045 ZX$=INKEY$:IF
  ZX$="" "THEN 10045 ELSE IF
  ZX$="-"THEN
  ZZ$="-":PRINTZX$;:GOTO
  10051 ELSE GOTO 10051
10050 ZX$=INKEY$: IF
  ZX$="" " THEN 10050
10051 IF ZX$=CR$ THEN
  ZZ=VAL (ZZ$):RETURN
10060 IF (ZX$»"9") OR
  (ZX$«".") OR (ZX$="/")THEN
  10050 ELSE
  ZZ$=ZZ$+ZX$:PRINTZX$;:
  GOTO 10050
```

This looks clumsy, but isn't really — it can keep up with fairly fast typing. Don't use it in your new word processor though.

Tricks such as this have other applications too. The clumsy user who hits shift-backspace will be surprised to find all he has typed disappear. A simple problem — use a routine like those outlined to filter out the keys which are not to be acted on.

Another help to the user is to force correct input by giving the proper format on the screen, and ignoring anything that doesn't fit. As an example, here is a subroutine to inpute a date.

The format ../../.. is displayed, and as characters are input, they fill up the spaces. The day, and month, figures are checked for reasonableness as soon as they are complete. The disadvantage — single digit days and months must be given a leading zero. It isn't a lot more work to program around this; you need only check for a carriage return, or arrow key or whatever you choose to trigger a move to the next field.

```
10 SCRN = 128 ' where it all
   happens.
20 CLS
30 FMT$="../../.."
40 GOSUB 10000
50 END
55'-----------------------------------
9997 '
9998 ' date input subroutine.
9999 '
10000 PRINT @ SCRN,
   FMT$;:PRINT@SCRN," ";
   ' print format and go back to
   beginning of it.
10002 ZZ$=" ":GOSUB
   10030:DAY = VAL(ZZ$)
10004 IF (DAY » 0) AND (DAY «
32) THEN GOTO 10008 ' skip er-
ror message.
10006 PRINT @ SCRN+64,"ER-
ROR: incorrect day. Press any
key to continue.": GOSUB 10040:
PRINT @ SCRN+64,
CHR$(31);:GOTO 10000
10008 PRINT @ SCR+3," "; '
move to month field
10010 ZZ$=" ":GOSUB
10030:MTH = VAL(ZZ$)
10012 IF (MTH » 1) AND (MTH »
13) THEN GOTO 10016 ' skip
error
10014 PRINT @ SCRN+64, "ER-
ROR: incorrect month. Press any
key to continue.": GOSUB
10040: PRINT @ SCRN+64,
CHR$(31);:GOTO 10008
10016 PRINT @ SCRN+6,
" ";:ZZ$=" ":GOSUB 10030
10018 RETURN ' to main program
calling 10000
10019 ' next subroutine sets one
digit (only!) and adds to ZZ$
10020 ZX$=INKEY$: IF (ZX$ «
"0") OR (ZX$» "9") THEN
10020 ELSE
ZZ$=ZZ$+ZX$:RETURN
10029 ' this subroutine sets two
digits, echos them.
10030 FOR I=1 TO 2:GOSUB
10020:
PRINTZX$;:NEXT:RETURN
10040 IF INKEY$= " " THEN
10040 ELSE RETURN : pause until
key pressed.
```

Of course, if you want to process or store the date, you need to pick up the day, month and year as they are input.

This sounds a lot of trouble, and not very interesting. But it isn't really, and the results are well worthwhile. Besides, you need only write the code once to have it for use in any number of programs.

This little demonstration looks fairly good — and it will look even better when you add a flashing cursor to it!

Sinclair book a good buy

"Your Timex Sinclair 1000 And ZX81."
Published by Sybex Publications, Author, Douglas Hergart. Price $12.50. Reviewed by Tony Lewis.

At first glance this book looks like a rewrite of the ZX81 manual. Further inspection reveals that it is a lot more. Used in conjunction with the ZX81 manual it would be a very good guide to the beginner on the Timex 1000 or Sinclair ZX81.

There are five chapters and two appendices. Chapter one, "The Cast of Characters," goes through the initial setting up of the machine and what each part does and what your role is.

Chapter two, "The First Act: Enter Your Program," deals with the computer keyboard and the BASIC language used.

Chapter three, "The Plot Thickens: A Short, Graphic, Course in BASIC," looks at the graphics capabilities of the Timex 1000 and the commands needed to use them.

Chapter four, "Take Five: Numbers on Your Computer," explains how the ZX81 or Timex 1000 can be used as a calculator and how to generate bar graphs.

Chapter five, "Words, Words, Words: Strings and String Functions on Your Computer," completes the book by explaining the use of strings and how you can "slice" them.

Programs are used throughout the book to illustrate what is being covered in each chapter. Appendix A gives a list of the BASIC vocabulary and what each word means.

Appedix B gives a list of error codes used by the Timex 1000 and Sinclair ZX81.

Most programs in the book will run on the Sinclair ZX81 but since the Timex 1000 has 2K of memory instead of the IK of its English counterparts, there are some programs that require 2K.

In summary, in conjunction with the Sinclair manual this book could be very useful to the beginner as it covers some areas that the manual overlooked. Value for money it has to be a good buy.

Useful Pascal work

"From BASIC to Pascal" by Ronald W. Anderson. Published by TAB Books Inc. 310 Pages. $21.95. Reviewed by Gerrit Bahlman.

If you are an experienced BASIC programmer who needs to be convinced that there is something in the structured beast, Pascal, then this book may be what you are looking for. By no means is it a beginners guide to Pascal. It offers a solid introduction which assumes a sophisticated comprehension of programming jargon and makes little concession to the novice.

Notions such as the distinction between value and variable parameters are not explained in a simple way, utilising examples and diagrams — it is explained in a textual fashion that would leave the novice stranded.

The book provides a very good comparative study between the two languages by programs presented in BASIC then Pascal, with explanations of their differences. The old much used recursion example of "Factorials" is presented as is a non-recursive version of the same exercise. While interesting, this is the only example of recursion cited and once more fails to satisfy the need for more appropriate examples of this difficult concept. This is particularly true of the experienced BASIC programmer who has developed a mental attitude to non-recursive solution of problems which make recursion approaches difficult to grasp.

The most attractive feature of

Continued page 35

PIPS III update — memory file pages

By Peter Hyde

Over the past few months, a number of SORD M23 users have bought and are using PIPS-III, the latest release of SORD's Personal Information Processing System. In that time some discoveries have been made which will make the use of PIPS-III easier, and increase the power of the system.

The first of these discoveries relates to getting the full use of the 128K of memory available in the M23. When PIPS-III is loaded, most of this is occupied by the PIPS program itself and the memory work areas of PIPS which you know as the Master Buffer, Sub Buffer and Figure Buffer.

The disk drives provide the permanent storage areas known as the Master File and Sub File. However, there is a temporary storage area known as the Memory File, which can hold three PIPS-III pages. This large storage area in memory is useful when you want to work with several pages of data without having to access the disk.

Transfer of the contents of these pages to and from the Master Buffer is almost instantaneous. Thus you can save a second or two by saving pages in the Memory File instead of the Master File or Sub File.

Note: Because the pages are stored in Memory, their contents will be lost if you turn off the power. If you exit from PIPS, and then re-enter without first powering off, the memory pages (like the Master Buffer) will be intact, except for the first one.

Now let's find out how to reference the three extra pages. This is by means of the P (Put) and G (Get) commands used for saving and retrieving pages on disk.

Just as the statement: P;1 saves the Master Buffer content in Master File page one, so: P;C1 saves the Master Buffer in the first Memory File. The codes C2 and C3 refer to the other two pages available.

To retrieve the second page enter: G;C2 and press RETURN. Whatever is in that page will instantly appear in the Master Buffer on the display screen.

These memory pages cannot be referenced by PIPS-III commands such as SORT, CS (Conditional Search), CAL (Calculate), L (List to Printer) or UPD (Update). These commands, and a few others, allow you to specify a range of pages on which the command will operate (e.g. L;S1,8 prints Sub-Files one to eight inclusive). However, you cannot specify SORT:C3 . . . or CS:C1,2 . . .

The reason for this is that some of these commands actually use one or more of the Memory File pages for their own operations. For example, CS, UPD and CA use the first few lines of C1 for storing the search conditions. SORT sometimes uses C2. And C3 will contain any automatic program you are executing (or have recently finished executing).

This imposes a limitation on your use of the Memory File. Treat the pages as a purely temporary storage area (like the Sub-Buffer) and save them on disk before you execute a command which may use them. Above all, DON'T use C3 in an automatic program!

NOTE TO PIPS-II USERS: Yes, you have Memory File pages as well — 7 of them! They are accessed by the commands, "PCF" and "GCF", and do not require "C" prefixes on the page number, e.g. PCF;6.

Now that you have discovered the memory Files, there is one immediate application for them.

Many users have asked: "Is it possible to have PIPS remember my commands as I type them in so that I do not have to re-type them when creating an automatic program?"

Short answer: Yes. Memory page C3 can be used to log all your commands as you type them. You can then edit them (using the ED command) and finally register your new Automatic Program.

Use the following command sequence:

O; TITLE	This creates a blank page in the Master Buffer, with a title of your choice.
<ESC>	Press<ESC>key to end the O command.
P;C3	Save the blank page in C3.
SET; LOG=ON	Activate logging of your commands.
Issue any commands you wish to use in your programs, testing them as you go	All commands are remembered.
SET; LOG=OFF	De-activate logging.
G;C3	Get the list of commands.
ED	Edit your program. Remember to put STOP on the end, and ensure you delete any mistakes you made in keying.
AS;R;page number	Register your new program.

PIPS-III is designed as a full system to give you the means to do all normal business tasks as easily as possible. Automatic Programs extend the power of PIPS so that you can automate the jobs that you do. However, some task may be too complex for PIPS-III to handle easily, or they may have requirements for speed or communications which PIPS cannot handle by itself.

A special version of Basic called DBASIC-III has been created to get over this obstacle. With DBASIC-III (the "D" stands for "Docking") you can write programs in normal BASIC which process the data in PIPS pages in any manner you choose.

Furthermore, you can compile your DBASIC program, store it on the PIPS program disk, and call it up from within PIPS-III using the command: DK#PROG where PROG is the name of your program.

Thus you can write your own custom routines to do such jobs as communications and data conversion to/from PIPS-III pages. Anything you can do in SORD BASIC can now be done from within PIPS-III itself.

Getting loops straight

Basic BASIC 6

By GORDON FINDLAY

Continuing a series on BASIC for complete beginners.

Loops are so common in programming that almost all programming languages have special statements for coding them rapidly and easily. BASIC is no exception, and this installment will look at the FOR statement.

Here is a simple program with a loop. Type it in, remembering to make any changes your machine expects (putting in LET, for example.):

```
10 X = 1
20 Y = X * X
30 PRINT X; " ";Y
40 X = X +1
50 IF X<= 20 THEN GOTO 20
60 END
```

Let's analyse it: lines 20 and 30 do the real work — between them they calculate X * X; and print them out, separated by spaces. Line 10 makes this happen with X having a value of 1 at first; line 50 makes it happen again and again until X becomes 20; and line 40 makes sure the value of X increases by 1 each time. Lines 10, 40, and 50 together make up the loop.

Every loop has a "first" value, a "last" value, and an "increment", or step. In our program above, the "first" value is 1, the "last" value is 20, and the "increment" or step is 1. Try changing one or two of them, in lines 10, 40 or 50.

There is a neater way: BASIC provides the FOR statement. Modify the program to look like this:

```
10 FOR X = 1 TO 20 STEP 1
20 Y = X * X
30 PRINT X; " ";Y
40 NEXT X
60 END
```

The FOR statement in line 10 has the first, last and stepping values, and the NEXT X statement shows where the loop ends. These two act as a pair of brakes, as it were, enclosing the lines in the program which are repeated.

In most machines, the STEP 1 can be left out if the stepping value really is 1. So line 10 could become:

```
10 FOR X = 1 TO 20
```

Even if this is so, you must include the STEP phrase if the increase is by some other value. Change line 10 to read

```
10 FOR X = 1 TO 20 STEP 0.5
```

Now X is increased by 0.5 each time round the loop, so more values are produced.

Here is another program to try. Work out just what is produced, then try with a machine and see if you were right:

```
10 FOR Y = 6 TO 25 STEP 0.25
20 Z = 2 * Y
30 W = 3 * Y
40 PRINT Z, W
50 NEXT Y
60 END
```

There are a few things to remember about FOR-loops to prevent disasters. Actually, these restrictions are true for any sort of loop, however constructed. It is just that they are easier to express and more obviously dangerous in this case.

Firstly, don't miss out the NEXT statement. If you do, the rest of the program will be included in the first interaction (first time round the loop), and then the program is finished.

Secondly, name the correct variable in the NEXT statement. Some of the better versions of BASIC can make do without the variable at all, but even then you must be careful not to include the wrong one.

It is often sensible to jump OUT of a FOR-loop, but it isn't very sensible to jump IN. If you do jump in, what is the value of the looping variable to be? Now I know it is

BEGINNERS

possible to fix things to make jumping in work, but there is always a better way.

FOR-loops are often found nested. But if they are, the inner and outer loops must never cross. If you think about it, you should be able to see why — after all, what would the value of J be in line 70 in this incorrect skeleton:

```
10 FOR J = 1 TO 10
20 . . . . . .
30 FOR K = 10 TO 20
40 . . . . . .
50 . . . . . .
60 NEXT J
70 . . . . . .
80 NEXT K
```

The program needs a value of K in line 70, but since K depends on J and line 65 is outside the J-loop, K cannot have a value. This is only one of the sort of things which can happen with crossed loops, and they are outlawed completely.

One sure way to prevent nested FOR-loops from crossing is to be sure the NEXT statements come in the reverse order to their matching FORS.

Here is an example of a program with correctly nested loops — all it does is prints out the multiplication tables:

```
10 FOR I = 1 TO 10
20 FOR J = 1 TO 10
30 K=I*J
40 PRINT I, "x";J" = ";K
50 NEXT J
60 NEXT I
70 END
```

See how the J-loop is totally inside the I-loop, not crossing it. Now a job for you: modify this program to print a multiplication table in the usual square form.

From page 21

transfer time differences cited would be far less apparent. The lesson here is that your peripherals and network should be on the same speed plane. A super-fast network running slow peripherals is wasted money.

6 Can I change the machines I use on my network

The user's dream is a network that any microcomputer can attach to. The reality is that commercial pressures have encouraged the manufacturers to specialise their

Continued page 35

Equation: Winning program

"Equation" is for a Sinclair ZX81 with 16K RAM, and unexpanded ROM.

This is the winning program in "Bits & Bytes" competition No. 1. It is by Wayne Dobson, of Karamu High School, Hastings, and won him a Sinclair ZX81 supplied by David Reid Electronics.

When "Equation" is run the name of the program and the writer's name are displayed. Then the program goes into an example showing movement keys, how to play, and the point scoring. That goes away and the computer simply displays in the middle of the screen, "Push G to PLAY". If G is not pushed then after a while that clears and an example of the high score being written up is shown. Another "Push G to PLAY" is displayed and if G is not pushed then the cycle starts again.

Note, however, that if there is a high score it is displayed (just before the program's name) with the highest scorer and the holder's age.

If G is pushed then the screen clears and you are asked to push either N for novice, I for Intermediate, or E for expert. Novice has a fixed height for the equations to come from and a slightly longer time to answer the question (lines 2130 to 2140). In intermediate, the equations can come from 4 different heights (line 3009) and the player has a shorter period to answer the equation. In expert, the player has a very short time to answer the equation and it can come from eight different heights. To stop people from working out a system the answer ship (your ship which is controllable) comes from different places.

The game section — when playing the game the question will appear (addition or subtraction) and top. The player must then hold down the key (0 to 9) which answers the question until the scores at the top change and the answer is written up where it is to be displayed. Then the question starts moving and the player by moving the answer ship (an A at the bottom of the screen) using keys 5 (1 TO THE LEFT), 8 (1 TO THE RIGHT) and 9 (2 TO THE RIGHT) must position himself or herself ahead of the question and by using key 0 (zero) must release the answer from the ship in time for it to move up the screen and connect (if possible) to the end of the question (scoring shows points and misses).

Five misses is the end of the game and the highest scorer is allowed to put your name and age up.

N.B. failure to answer the equation correction or push answer up at ship results a miss also keys must be securely held down, not just taped.

Wayne has suggested the following amendments to his entry:

Add —
5003 PRINT AT 0,21;C;AT0,21;" "
5005 PRINT AT 0,21;C;AT0,21;" "
Change lines —
2167 TO 2169 and replace 2167 with the original 2169
3167 TO 3169 and replace 3167 with the original 3169
4167 TO 4169 and replace 4167 with the original 4169

The program
1 & 10 Copyright, name and writers name, date prog was written.
30 sets high-score to zero.
40 checks to see if there is a high score, if not jump to line 160.
42 to 150 Displays high score, high scores holder, holders age.
160 to 290 Displays Prog's name, writer and month & year written.
300 to 960 Draws a mock screen and game as in the pictures of a novice game.
970 to 1020 asks if you want to play.
1030 to 1300 shows you what to do if you get the high score.
1310 to 1340 asks if you want to play.
1350 Loops back to 40.
1400 to 1630 asks you to choose a class (N, I, or E).
2000 to 2460 is the novice section.
3000 to 3460 is the intermediate section.
4000 to 4460 is the expert section.
5000 to 5160 is for a high score to be written up.

Variables
A B$ as a decimal number
B Score
C Hi-Score
E Score this turn
F Height of equation
G Used to play Game
H Column of answer ship
I & J Used for loops
K To give a number deciding if the equation is + or −
L High score holders age
M First factor of equation
N Second factor of equation
O Sum of equation
Q Number of misses
T Movement of equation in example
Z Movement of answer in example
A$ 31 (1 line) of spaces
B$ Used for answer to equation
C$ Hi Score holders name
D$ "4 +5=" for example
E$ Numbers for "A" movement in example
F$ Number 9 for example

Difficult lines to read
2450 print at 8,0; A$; at 18,H; " " — one space.
2460 go to 2005.
3450 print at 8,0; A$; at 18,H; " " — one space.
3460 go to 3005.
4450 print at 8,0; A$; at, H; " " — one space.
4460 go to 4005.
1460 let O=0.

```
   1 REM COPYRIGHT [illegible]
  10 REM **EQUATION**BY W R
DOBSON NOV 1982
  20 REM HI-SCORE
  30 LET C=0
  40 IF C=0 THEN GOTO 160
  42 PRINT AT 0,10;" [illegible]"
  43 PRINT AT 1,10;" [illegible]"
  44 PRINT AT 2,10;" [illegible]"
  45 PRINT AT 3,10;" [illegible]"
  46 PRINT AT 4,12;" [illegible]"
  50 PRINT AT 7,2;" [illegible]"
  60 PRINT AT 8,2;" [illegible]"
  70 PRINT AT 9,2;" [illegible]"
  80 PRINT AT 10,2;" [illegible]"
  90 PRINT AT 11,2;" [illegible]"
 100 PRINT AT 12,2;C;" POINTS"
 110 PRINT AT 14,2;"HELD BY"
 120 PRINT AT 16,2;A$
 130 PRINT AT 18,2;"AGE ";L
 140 FOR I=0 TO 150
 150 NEXT I
 160 REM NAME
 170 CLS
 180 PRINT AT 3,9;"EQUATIONEQUAT
I"
 190 PRINT TAB 9;"Q";TAB 22;"O"
 200 PRINT TAB 9;"U";TAB 22;"N"
 210 PRINT TAB 9;"A";TAB 22;"O"
 220 PRINT TAB 9;"T";TAB 22;"I"
 230 PRINT TAB 9;"I";TAB 22;"T"
 240 PRINT TAB 9;"ONEQUATIONEQUA
"
 250 PRINT AT 3,10;"EQUATION"
 260 PRINT AT 12,7;"BY WAYNE R D
OBSON"
 270 PRINT TAB 11;"NOVEMBER 1982

 280 FOR I=0 TO 300
 290 NEXT I
 300 REM EXAMPLE
 310 CLS
 320 PRINT "SCORE=0";AT 0,11;"HI
-SCORE=";C;AT 0,24;"MISSES=0"
 330 PRINT AT 1,0;"SCORE THIS TU
RN=0";AT 1,20;"ANSWER="
 340 PRINT AT 21,0;"EQUATION BY
W R DOBSON 1982"
 350 PRINT AT 20,0;"LEFT=5";AT 2
0,19;"RIGHT 1=8,2=9"
 360 PRINT AT 6,7;"PUSH G TO PLA
Y"
 370 PRINT TAB 9;"WHEN ASKED"
 380 LET D$="4+5="
 390 PRINT AT 8,0;D$
 400 FOR I=0 TO 2
 410 NEXT I
 420 PRINT AT 1,27;9
 430 FOR I=0 TO 5
 440 NEXT I
 450 LET H=10
 460 PRINT AT 18,H;" A "
 470 FOR I=1 TO 4
 480 LET E$="9558"
 490 IF E$(I)="9" THEN LET H=H+2
 500 IF E$(I)="5" THEN LET H=H-1
 505 IF E$(I)="8" THEN LET H=H+1
 510 PRINT AT 8,I;D$
 520 PRINT AT 18,H;" A "
 530 FOR J=0 TO 6
 540 NEXT J
 550 NEXT I
 560 LET Z=17
 570 LET T=9
```

```
580 LET F$="0"
590 IF Z<17 THEN PRINT AT Z+1,H+2;" "
600 PRINT AT Z,H+2;F$
610 PRINT AT 0,F;O$
620 LET T=T+1
630 FOR I=0 TO 8
640 NEXT J
650 LET Z=Z-1
660 IF T<=14 THEN GOTO 590
670 PRINT AT 11,3;"5 POINTS"
680 FOR I=0 TO 30
690 NEXT I
700 PRINT AT 8,14;" 4+20 ";AT 8,14;" "
710 PRINT AT 11,3;"4 POINTS"
720 FOR I=0 TO 30
730 NEXT I
740 PRINT AT 8,14;" 4+9= ";AT 8,14;" "
750 PRINT AT 11,3;"3 POINTS"
760 FOR I=0 TO 30
770 NEXT I
780 PRINT AT 8,14;" 495= ";AT 8,14;" "
790 PRINT AT 11,3;"2 POINTS"
800 FOR I=0 TO 30
810 NEXT I
820 PRINT AT 8,14;" 9+5= ";AT 8,14;" "
830 PRINT AT 11,3;"1 POINT "
840 FOR I=0 TO 30
850 NEXT I
860 PRINT AT 8,11;"91+5= ";AT 8,14;" "
870 PRINT AT 11,3;"0 POINTS"
880 FOR I=0 TO 30
890 NEXT I
900 PRINT AT 8,10;"9 4+5= ";AT 8,13;" "
910 PRINT AT 11,3;"MISS     "
920 FOR I=0 TO 30
930 NEXT I
940 PRINT AT 8,10;"  4+5= 9";AT 8,13;" "
950 FOR I=0 TO 300
960 NEXT I
970 REM PUSH 0 TO PLAY
980 CLS
990 PRINT AT 11,7;"PUSH 0 TO PL"
1000 FOR I=0 TO 300
1010 IF INKEY$="0" THEN GOTO 140
1020 NEXT I
1030 REM EXAMPLE (HIGH SCORE)
1040 CLS
1050 PRINT "EXAMPLE"
1060 PRINT AT 0,5;"CONGRATULATIO"
1070 PRINT AT 7,2;"YOU HAVE THE HIGH SCORE"
1080 PRINT AT 9,3;"ENTER YOUR"
1090 PRINT AT 11,5;"NAME"
1100 FOR X=0 TO 300
1110 PRINT AT 21,0;"W";
1120 LET C$="AYNE DOBSON"
1130 FOR I=1 TO 11
1140 FOR J=0 TO 5
1150 NEXT J
1160 PRINT C$(I);
1170 NEXT I
1180 PRINT AT 21,0;"
1190 PRINT AT 11,6;"WAYNE DOBSON"
1200 PRINT AT 13,2;"AGE"
1210 PRINT AT 21,0;1
1220 FOR I=0 TO 9
1230 NEXT I
1240 PRINT AT 21,0;15
1250 FOR I=0 TO 9
1260 NEXT I
1270 PRINT AT 21,0;"  "
1280 PRINT AT 13,6;"15"
1290 FOR I=0 TO 200
1300 NEXT I
1310 REM PUSH 0 TO PLAY
1320 FOR I=0 TO 300
1330 IF INKEY$="0" THEN GOTO 1400
1340 NEXT I
1350 GOTO 40
1400 REM ***GAME***
1410 CLS
1415 LET MO=0
1420 LET A=0
1430 LET B=0
1450 LET E=0
1460 LET Q=0
1470 LET G=0
1480 LET H=0
1490 LET I=0
1500 LET J=0
1510 LET K=0
1530 LET M=0
1540 LET N=0
1550 LET O=0
1560 PRINT "SCORE=";AT 0,11;"HI-SCORE=";AT 0,24;"MISSES="
1570 PRINT AT 1,0;"SCORE THIS TURN=";AT 1,20;"ANSWER="
1580 PRINT AT 21,0;"WARNING YOU HAVE ONLY ONE CHANCE"
1590 PRINT AT 0,6;"CHOOSE YOUR CLASS"
1600 PRINT TAB 8;"N-NOVICE";TAB 8;"I-INTERMEDIATE";TAB 8;"E-EXPERT"
1603 IF INKEY$="N" OR INKEY$="I" OR INKEY$="E" THEN GOTO 1605
1604 GOTO 1603
1605 LET A$="
1607 PRINT AT 5,0;A$;A$;A$;A$
1610 IF INKEY$="N" THEN GOTO 2000
1620 IF INKEY$="I" THEN GOTO 3000
1630 IF INKEY$="E" THEN GOTO 4000
```

```
2000 REM ***NOVICE***
2005 LET E=0
2006 PRINT AT 5,0;A$
2007 PRINT AT 1,27;" ";AT 10,H;" "
2010 LET K=INT (RND*2)
2020 LET M=INT (RND*10)
2030 LET N=INT (RND*10)
2031 PRINT AT 0,6;B;AT 0,21;C;AT 0,31;O;AT 1,16;E
2040 IF K=0 THEN LET O=M+N
2050 IF K=0 AND O>=10 THEN GOTO 2020
2060 IF K=1 THEN LET O=M-N
2070 IF K=1 AND O<=0 THEN GOTO 2020
2080 LET H=INT (RND*15)+10
2090 PRINT AT 10,H;"A"
2100 LET Z=1
2110 IF K=0 THEN PRINT AT 8,Z;M;"+";N;"=";
2120 IF K=1 THEN PRINT AT 8,Z;M;"-";N;"=";
2130 FOR I=1 TO 30
2135 LET B$=INKEY$
2140 IF B$="" THEN NEXT I
2150 LET A=CODE B$-28
2160 IF A=0 THEN LET E=E+1
2165 LET B=B+E
2166 IF B>C THEN LET C=B
2167 IF NOT A=O THEN LET Q=Q+1
2168 IF B>C THEN LET G=1
2169 PRINT AT 1,16;E;AT 0,6;B;AT 0,21;C;AT 0,31;Q
2170 IF Q>=5 THEN GOTO 5000
2175 IF NOT A=O THEN GOTO 2450
2180 PRINT AT 1,27;A
2190 FOR Z=1 TO 27
2200 IF K=0 THEN PRINT AT 8,Z;" ";M;"+";N;"="
2210 IF K=1 THEN PRINT AT 8,Z;" ";M;"-";N;"="
2215 LET G=H
2220 LET H=H+(INKEY$="8")+(INKEY$="9")+(INKEY$="8")-(INKEY$="5")
2221 IF H>=31 THEN LET H=31
2222 IF H<=0 THEN LET H=0
2225 IF NOT H=G THEN LET MO=1
2230 IF INKEY$="0" THEN LET A=10
2240 IF A=10 THEN GOTO 2300
2250 PRINT AT 10,H;"A"
2260 IF MO=1 THEN PRINT AT 10,G;" "
2270 NEXT Z
2280 LET Q=Q+1
2290 GOTO 2000
2300 LET HT=17
2305 IF K=0 THEN PRINT AT 8,Z;" ";M;"+";N;"="
2310 IF K=1 THEN PRINT AT 8,Z;" ";M;"-";N;"="
2320 LET Z=Z+1
2330 IF HT<17 THEN PRINT AT HT+1,H;" "
2340 PRINT AT HT,H;O
2350 LET HT=HT-1
2360 IF HT>=8 THEN GOTO 2305
2370 IF HT=7 AND H=Z+4 THEN LET E=E+5
2380 IF HT=7 AND H=Z+3 THEN LET E=E+4
2390 IF HT=7 AND H=Z+2 THEN LET E=E+3
2400 IF HT=7 AND H=Z+1 THEN LET E=E+2
2410 IF HT=7 AND H=Z THEN LET E=E+1
2420 IF HT=7 AND H<=Z-2 THEN LET Q=Q+1
2430 IF HT=7 AND H>=Z+5 THEN LET Q=Q+1
2440 LET B=B+E
2443 PRINT AT 1,16;E;AT 0,6;B
2444 IF B>C THEN LET G=1
2445 IF B>C THEN LET C=B
2446 IF Q>=5 THEN GOTO 5000
2447 PRINT AT 0,21;C
2450 PRINT AT 5,0;A$;AT 10,H;" "
2460 GOTO 2005
3000 REM ***INTERMEDIATE***
3005 LET E=0
3006 PRINT AT 5,0;A$;A$;A$;A$;A$
3007 PRINT AT 1,27;" ";AT 10,H;" "
3009 LET F=INT (RND*4)+7
3010 LET K=INT (RND*2)
3020 LET M=INT (RND*10)
3030 LET N=INT (RND*10)
3040 IF K=0 THEN LET O=M+N
3050 IF K=0 AND O>=10 THEN GOTO 3020
3060 IF K=1 THEN LET O=M-N
3070 IF K=1 AND O<=0 THEN GOTO 3020
3080 LET H=INT (RND*15)+10
3090 PRINT AT 10,H;"A"
3100 LET Z=1
3110 IF K=0 THEN PRINT AT F,Z;M;"+";N;"=";
3120 IF K=1 THEN PRINT AT F,Z;M;"-";N;"=";
3130 FOR I=1 TO 20
3135 LET B$=INKEY$
3140 IF B$="" THEN NEXT I
3150 LET A=CODE B$-28
3160 IF A=O THEN LET E=E+1
3165 LET B=B+E
3166 IF B>C THEN LET C=B
3167 IF NOT A=O THEN LET Q=Q+1
3168 IF B>C THEN LET G=1
3169 PRINT AT 1,16;E;AT 0,6;B;AT 0,21;C;AT 0,31;Q
3170 IF Q>=5 THEN GOTO 5000
3175 IF NOT A=O THEN GOTO 3450
3180 PRINT AT 1,27;A
3190 FOR Z=1 TO 27
3200 IF K=0 THEN PRINT AT F,Z;" ";M;"+";N;"="
3210 IF K=1 THEN PRINT AT F,Z;" ";M;"-";N;"="
```

```
3215 LET G=H
3220 LET H=H+(INKEY$="8")+(INKEY$="9")+(INKEY$="8")-(INKEY$="5")
3221 IF H>=31 THEN LET H=31
3222 IF H<=0 THEN LET H=0
3225 IF NOT H=G THEN LET MO=1
3230 IF INKEY$="0" THEN LET A=10
3240 IF A=10 THEN GOTO 3300
3250 PRINT AT 10,H;"A"
3260 IF MO=1 THEN PRINT AT 10,G;" "
3270 NEXT Z
3280 LET Q=Q+1
3290 GOTO 3000
3300 LET HT=17
3305 IF K=0 THEN PRINT AT F,Z;" ";M;"+";N;"="
3310 IF K=1 THEN PRINT AT F,Z;" ";M;"-";N;"="
3320 LET Z=Z+1
3330 IF HT<17 THEN PRINT AT HT+1,H;" "
3340 PRINT AT HT,H;O
3350 LET HT=HT-1
3360 IF HT>=F THEN GOTO 3305
3370 IF HT=F-1 AND H=Z+4 THEN LET E=E+5
3380 IF HT=F-1 AND H=Z+3 THEN LET E=E+4
3390 IF HT=F-1 AND H=Z+2 THEN LET E=E+3
3400 IF HT=F-1 AND H=Z+1 THEN LET E=E+2
3410 IF HT=F-1 AND H=Z THEN LET E=E+1
3420 IF HT=F-1 AND H<=Z-2 THEN LET Q=Q+1
3430 IF HT=F-1 AND H>=Z+5 THEN LET Q=Q+1
3440 LET B=B+E
3443 PRINT AT 1,16;E;AT 0,6;B
3444 IF B>C THEN LET G=1
3445 IF B>C THEN LET C=B
3446 IF Q>=5 THEN GOTO 5000
3447 PRINT AT 0,21;C
3450 PRINT AT 5,0;A$;AT 10,H;" "
3460 GOTO 3005
4000 REM ***EXPERT***
4005 LET E=0
4006 PRINT AT 5,0;A$;A$;A$;A$;A$;A$;A$;A$
4007 PRINT AT 1,27;" ";AT 10,H;" "
4009 LET F=INT (RND*8)+9
4010 LET K=INT (RND*2)
4020 LET M=INT (RND*10)
4030 LET N=INT (RND*10)
4040 IF K=0 THEN LET O=M+N
4050 IF K=0 AND O>=10 THEN GOTO 4020
4060 IF K=1 THEN LET O=M-N
4070 IF K=1 AND O<=0 THEN GOTO 4020
4080 LET H=INT (RND*15)+10
4090 PRINT AT 18,H;"A"
4100 LET Z=1
4110 IF K=0 THEN PRINT AT F,Z;M;"+";N;"=";
4120 IF K=1 THEN PRINT AT F,Z;M;"-";N;"=";
4130 FOR I=1 TO 15
4135 LET B$=INKEY$
4140 IF B$="" THEN NEXT I
4150 LET A=CODE B$-28
4160 IF A=O THEN LET E=E+1
4165 LET B=B+E
4166 IF B>C THEN LET C=B
4167 IF NOT A=O THEN LET Q=Q+1
4168 IF B>C THEN LET G=1
4169 PRINT AT 1,16;E;AT 0,6;B;AT 0,21;C;AT 0,31;Q
4170 IF Q>=5 THEN GOTO 5000
4175 IF NOT A=O THEN GOTO 4450
4180 PRINT AT 1,27;A
4190 FOR Z=1 TO 27
4200 IF K=0 THEN PRINT AT F,Z;" ";M;"+";N;"="
4210 IF K=1 THEN PRINT AT F,Z;" ";M;"-";N;"="
4215 LET G=H
4220 LET H=H+(INKEY$="8")+(INKEY$="9")+(INKEY$="9")-(INKEY$="5")
4221 IF H>=31 THEN LET H=31
4222 IF H<=0 THEN LET H=0
4225 IF NOT H=G THEN LET MO=1
4230 IF INKEY$="0" THEN LET A=10
4240 IF A=10 THEN GOTO 4300
4250 PRINT AT 18,H;"A"
4260 IF MO=1 THEN PRINT AT 18,G;" "
4270 NEXT Z
4280 LET Q=Q+1
4290 GOTO 4000
4300 LET HT=17
4305 IF K=0 THEN PRINT AT F,Z;" ";M;"+";N;"="
4310 IF K=1 THEN PRINT AT F,Z;" ";M;"-";N;"="
4320 LET Z=Z+1
4330 IF HT<17 THEN PRINT AT HT+1,H;" "
4340 PRINT AT HT,H;O
4350 LET HT=HT-1
4360 IF HT>=F THEN GOTO 4305
4370 IF HT=F-1 AND H=Z+4 THEN LET E=E+5
4380 IF HT=F-1 AND H=Z+3 THEN LET E=E+4
4390 IF HT=F-1 AND H=Z+2 THEN LET E=E+3
4400 IF HT=F-1 AND H=Z+1 THEN LET E=E+2
4410 IF HT=F-1 AND H=Z THEN LET E=E+1
4420 IF HT=F-1 AND H<=Z-2 THEN LET Q=Q+1
4430 IF HT=F-1 AND H>=Z+5 THEN LET Q=Q+1
```

SINCLAIR

```
4440 LET B=B+B
4443 PRINT AT 1,16;E;AT 0,6;B
4444 IF B>C THEN LET C=B
4445 IF B>=5 THEN GOTO 5000
4447 PRINT AT 0,21;C
4450 PRINT AT 0,0;A$;AT 18,N;" "
4460 GOTO 4005
4990 REM ***NEW HIGH SCORE***
5000 FOR I=0 TO 100
5010 NEXT I
5020 CLS
5030 IF NOT G=1 THEN GOTO 40
5040 PRINT AT 0,5;"CONGRATULATIO
NS"
5050 PRINT AT 7,2;"YOU HAVE THE
HIGH SCORE"
5060 PRINT AT 9,2;"ENTER YOUR"
5070 PRINT AT 11,2;"NAME ";
5080 INPUT C$
5090 PRINT C$
5100 PRINT AT 13,2;"AGE"
5110 INPUT L
5120 PRINT AT 13,7;L
5130 FOR I=0 TO 60
5140 NEXT I
5150 CLS
5160 GOTO 40
```

BOOKS

From page 29

the book is the depth to which particular "standard" programming problems are developed. One of the familiar tasks allotted to the programming student is an analysis of sort rates using various methods. This text provides a comparison program which will appear like mana from heaven to a student required to produce such a scheme. The sorts examined are the bubble sort, (tried and slow!), insertion sort, insertion sort with binary search procedure, shell-metzner sort procedure and the quicksort procedure. The second point about these programs is that they are written so that the reader can utilise them as procedures in their own programs.

There are a number of other programs which may also be of use to a new Pascal programmer who has as yet not built up a library of routines which might be needed. Linked list procedures, a label print routine, character conversion routines, a simple data base with editing procedures, and even a pseudo-random number generator.

As a collector of esoteric routines that everyone writes as an exercise at one time or another I found the book immediately interesting, but, and there is almost alway a but . . .

The layout of the text, which is blocky and unattractive, the lack of space and diagrams combined with the newsprint/paperback quality of the materials used makes the book initially seem less than it actually is.

In summary, it is a book which does not patronise the reader and expects that the reader has a modicum of experience in programming and has probably undergone a series of formal programming exercises in BASIC. It presents a library of common procedures that can be incorporated into any programming work without undue modification. Its presentation is not attractive but this is misleading. Perhaps the final point I can make is that I am pleased to have a copy — it will be useful!

APPLE

A tip for Apple users

A reader, G. PORTENERS, of Paremata, offers this hint to Apple users

You may be courting trouble if you are using the RENUMBER program of APPLE II PLUS SYSTEM MASTER (DOS version 3.3 08/25/80). Try the following listing:

```
10 Q = A*20
20 R = B*20
30 END
```

Renumbering this by command & F = 5 results in:

```
5 Q = A*15
15 R = B*15
25 END
```

The factor 20 in lines 10 and 20 assumes the value of the renumbered line 20. If, however, you write the expressions with the numerical factor first, i.e. Q = 20*A and R = 20*B, all is well. I had this trouble in a program which used 240 HPLOT N, SIN(A) *80 + 79. On renumbering the original line 80 changed to 100 and so did factor 80 to produce SIN(A)*100. I have now changed the expression to 80*SIN(A) and have had no further trouble.

From page 32

networks. Having hooked you they have you. Having said that, there is an escape. Most machines can be made to converse with limited flexibility through serial ports (this is fine for occasional data transfer but limited in every other way). The light is that there are a few networks coming in to production which allow far greater flexibility, and this may be a significant purchasing factor. One network manufacturer, Omninet, was predicting a network capable of linking 23 different sorts of machine. There are limitations still in such an arrangement though.

- **Next month, Nick Smythe will look at networks which are available in New Zealand.**

COMPUTASHOP.

NZ's Widest Selection of Personal Computers monitors, printers, disk drives

*ZX-80 *Vic 20 *Hewlett Packard
*Atari *System 80 *BBC MICRO
*TRS80 Color Computer
* Osborne * Epson HX-20

We look forward to your visit. Write for Mail Orders.
Credit Cards Accepted

EINSTEIN SCIENTIFIC LIMITED
177 Willis Street, Wellington
PO Box 27138 Ph 851-055 Tx NZ3380

MicroAge MicroPro DISTRIBUTOR FOR N.Z.

* WordStar	$595	* SpellStar	$322	* MailMerge	$216
* DataStar	$440	* CalcStar	$380	* ReportStar	$410

IBM PC AND APPLE FORMATS AVAILABLE

MicroAge International (NZ) Ltd
P.O. Box 13-054
353-357 Hereford St,
Christchurch. Phone 891-109

---

computer point (N.Z.) Ltd

Christchurch
P.O. Box 25-091
phone 61-276

ALTOS BBC sirius Zilog

N.Z. Agents for ISE/PACTEL's MDBSIII Data Base Management System
Network specialists — ECONET — ETHERNET — ALTOSNET — CP/NET

KAYPRO II Portable Computer
COLOMBIA MPC 16 Bit Computer
Sole South Island distributors

TURNERS Ltd

P.O. Box 1021 Christchurch
Phone 794-820

AVM ELECTRONICS

FOR THE LATEST IN MICROCOMPUTERS
COMMODORE 64 & VIC20
ATARI
SINCLAIR

Lay Buy
Easy Terms
Trade-ins

EARN 14½% INTEREST WHILE YOU SAVE

149 HEREFORD STREET, CHRISTCHURCH PHONE 797-279
(Opp. Reserve Bank)

VIC 20

JAMES ELECTRONICS LTD
POLLEN ST
BOX 527
THAMES
Ph. 86-893

$795

Games tape 1. (needs 8k exp)
Snake-Maths Game-Ball in Bucket
Games tape 2. (no exp reqd)
Formula 1 - Concentration-Lane Game
$15 each p&p $1

Also available for VIC20 Payroll and Cashbook programs. Write for details.

COMPUTASHOP
COMPUTASHOP
COMPUTASHOP

Computashop is a low-cost way of reaching BITS & BYTES readers.

You can save up to 55 per cent on usual advertising rates.

Remember every BITS & BYTES reader is interested in computers — i.e. a guaranteed audience for your computer business or product.

Computashop,
BITS & BYTES,
Box 827,
Christchurch.

Please support BITS & BYTES advertisers

Beeb has Speed

By Pip Forer

Straining the deadlines and the editors' nerves to the limits just to get a glimpse of the first "Computer Programme" before this draft article gets sent off. It seems that the BBC and IBM have at least one thing in common (apart from each having three initials). That common thread is the realisation that the greatest barrier to computer adoption is the newcomer's nervousness in the face of technology. Charlie Chaplin and a red rose and Chris Serle's matter-of-fact probings provide the same ingredient: reassurance. Whether you prefer the slick symbols of one or the personal approach of the other is a matter of personal taste.

IBM also come into this column because this month we look briefly at the basic model B Beeb and ask just how good it is in its native tongue, BASIC. There are two approaches we could take here. One is to catlogue the merits and problems of BBC BASIC. The other is to benchmark the BBC computer against various other machines.

In general, BBC BASIC has been well received. Although its unique features will not transfer to other machines the combination of procedures, in-built assembler routines and powerful commands for handling Input/Output rather make up for that. VDU, OSWRCH and FX not only let you do things well within a BASIC program but make conversation with other machines that much easier.

BBC BASIC has been well covered in reviews elsewhere so, for the moment, further discussion of that aspect is put aside. Instead we will return to benchmarks and (indirectly) IBM. What is a benchmark? It is a timed series of operations on a computer, usually used to compare the processing power of one machine against another. The task defined within the benchmark forms the basis of comparison. That task can be very simple (and pretty stupid). A common first benchmark is to run a null loop, i.e.

```
10 for I = 1 to 1000
20 next I
```

For clarity you usually insert a print statement at the top (Ready, Steady, Go) and another at the end announcing the programme has finished. Then if you have a stopwatch you settle back and time the task. Of course, if you have a BBC you us the TIME command to do this for you. The command

```
5 TIME = 0
```

sets the clock counter to zero and

```
100 X =TIME:PRIN" "BENCHMARKED TIME": X/100:" SECONDS"
```

prints out the final lapsed time for the test

There are a few things to say about benchmarks. Firstly it is normal to run several on a single machine. This is because certain machines do some operations more quickly than others. A second test might be to add

```
15 GOSUB 1000
1000 RETURN
```

just to see how fast subroutine calls work. A machine fast at null loops may be slow at subroutine calls (and one fast at both may be slow with realy important instructions). The benchmarks usually therefore seek to test different aspects of a computer: setting variable values, partitioning string variables, doing trigonometric functions, doing normal maths operations. A favourite is to find the prime numbers between 1 and 1000. Since disk drive access time is frequently a large part of running time many benchmarks also involve disk access time by reading and writing randomly generated values to disk files.

Different publications publish and use different benchmarks in their evaluative reviews. Three I know of are those from "Australian Personal Computer," "Interace Age" and "Byte". How does the BBC Computer behave on some of these? The "APC" of December carried a list of 49 machines it had benchmarked. It used seven different programmes and averaged over these. They tested various aspects but had no disk access tests. The BBC came out as the second fastest. With a benchmark timing it ran through at 14.6 seconds. This compared with the slowest on that list, the Texas TI 99/4A with 78.2, the Apple II+ with 30.4 and the IBM PC with 17.6. The significant update since then has been the release of the Apple IIe with a reportedly much-enhanced clock speed. A comparative benchmark with this machine would be interesting.

Interested by the December benchmarks I ran the "Byte" benchmark used in a January, 1982, review of the IBM PC on a BBC machine. The figures are given below. They are for resident, non-compiled BASIC in either machine. The BBC machine figures are for a locally run benchmark, the IBM ones taken from "Byte," January, 1982, page 54.

Benchmark	IBM	BBC
1 Empty do loop	6.43	3.28
2 Division	16.80	23.80
3 Subroutine jump	12.4	5.34
4 Substring (MID$)	23.0	11.75
5 Prime Number	190.0	Out of space

I was unable to run the last benchmark fully since it requires a 7000 element array. The BBC machine has not enough memory to cope with this so we had to omit it. Nonetheless, the figures are quite interesting. Why do the differences emerge between machines, both here and in the "APC" table? First, factors include the efficiency of the BASIC and the clock speed of the processor in the machine. The BBC machine outdistances some other early 6502 machines simply because its clock speed is cranked up to 2 megahertz. Some machines suffer because the BASIC interpreter they carry has been poorly tailored to the processor they use. One thing that seems to emerge from most of the comparisons is that, at least for 8088 based 16-bit machines such

BBC

as the IBM PC you buy not for speed but for ease of handling a relatively large memory. Actual interpreted BASIC speeds suggest 8/16 bit hybrids such as the 8088-based machines are slower than a fast 8-bit machine but marginally faster than earlier 8-bit ones.

A word of caution, however. The "APC" table listed three 8088 based machines with reported speeds of 17.6, 20.9 and 24.8 respectively. That is each was roughly 20 per cent faster than the next. Running other benchmarks on the same machines locally suggested that the slowest and fastest were equivalent while the middle machine was about 25 per cent adrift of these two. The conclusion: benchmarks are useful only as rough guides, and at that as rough guides only to speed. For many users that may be no guide at all.

However, it is nice to have the power of a fast system if you need it. The BBC machine scores well on this for BASIC applications. However, one problem came straight to mind . . .it ran out of RAM. After all, benchmarks are designed to run on most machines and their demands should not be overly excessive (even if a 7000 number array is quite large). Available memory is the Achilles heel of the BBC machine as first released. To see how this arises and how it can be countered by additional paged memory or a second processor read on in May.

CLASSIFIEDS

FOR SALE CHALLENGER 1P, 8K with Cassette Tape Recorder, software and manuals, $500. Contact Mrs J. Wood, Cambridge High School, 5415.

WANTED Cash Flow, General Ledger accounting and farming software for SORD M100 Ace. Contact J.R. Stevens, 6A Euclid Street, New Plymouth. Phone 39-472.

FOR SALE

SURPLUS COMPUTER EQUIPMENT

IBM model 2770 RJE station with 2213/2 printer and 2502 card reader.
DEC RKO7 Disk Drives 28Mb (formatted) (two units)
DEC RK611 Disk Controller
DEC DZ11 A+B 16 line EIA controllers.

APS C30 6800 based microcomputers (3 units)

IBM and DEC equipment under maintenance agreements. Available now.

Contact: The Director
Computing Services Centre
Victoria University
Private Bag
Wellington
Tel 721-000 Ext 703

CUSTOM WRITTEN PROGRAMS:
FX702P, FX602P, System 80/TRS80, APPLEII, VIC20, and others. Other System 80 software also available. Write with stamped self addressed envelope for information to:
LINC
36 Bruce Rd.,
Glenfield,
Auckland 10.

ZX81 users: I would be interested to swap software (educational software if possible). T. Daniell, 8 John Street, Rangiora.

Exidy Sorcerer equipment. 48K computer, display-disk, S-100 expansion, ROM Pacs, D/A-A/D, tape recorder, EPROM eraser, etc. Will sell any or all to first reasonable offer or best offer by 30/4/83. Telephone Wellington 723-494.

ZX81 + 16K printer $350 Ph 796-606 (Wellington).

FOR SALE System 80 $800 ONO. 16K with sound lots of software. Phone Auckland 453-540.

WANTED: 80-micro back issues. 1980: Feb to May, 1981: Jan to June, Sept & Oct. Please sent issue nos & price to P. Clare, 8 Norway St, Kelburn, Wellington.

FOR SALE one 8K RAM pack, for the ZX81, includes reset button and power on LED $80.

ZX81 FOR SALE ONLY $169 with transformer and manual. Write 59 Wesley Street, Kaiapoi.

Casio FX-9000P for sale $900. As advertised in February 83 issue. S. Jasper, Hilltop, Ohai, Southland.

PROGRAMS for small business and personal computers written or modified to your specifications. Phone Paul Cull Christchurch 327-807.

VIC

From page 25

then takes them from there to use them on a first in, first out basis.

Our changes mean it doesn't use the buffer. It simply stores the Ns and Ys as we pushed them and leaves them there in the buffer. Normally the VIC stores 10 characters in the buffer and if these aren't used, it ignores any extras.

What we want to do is stop the VIC from storing any keys during the game. We can do this by poking the location that controls the number of characters stored. This location is 649. Try adding :POKE 649,0 to line 20 and ;POKE 649, 10 to line 220. That should solve the problem.

Now it's your turn to take over the process of refining the program. A few suggestions - try making the bat a different colour from the ball; add another bat and make it a two player game. You could either play tennis, second bat on the left, or squash, second bat at the same end.

If the game is too slow for you, change line 50 or perhaps allow for a variable to be input and then used in the timing loop. The possibilities are endless. Have fun!

CLUB CONTACTS

WHANGAREI COMPUTER GROUP: Tom Allan, 3 Maunu Rd, Whangarei. Phone 83-063 (w). Meets every second Wednesday of the month at Northland Community College.

NZ MICROCOMPUTER CLUB INC. P.O. Box 6210, Auckland. The monthly Meeting is held on the first Wednesday of each month at the VHF Clubrooms, Hazel Ave., Mt Roskill, from 7.30pm. Visitors are also welcome to the computer workshop in the clubrooms, 10am–5pm, on the Saturday following the above meeting.
The following user groups are part of the club. All meetings shown start 7.30pm at the VHF Clubroom.
Other active user groups within the club are:
APPLE, CP/M, DREAM 6800, SMALL BUSINESS, KIM, LNW, SORCERER, 1802 and 2650. They can all be contacted at club meetings or via NZ microcomputer Club, P.O. Box 6210, Auckland.
APPLE USERS' GROUP: Bruce Given, 12 Inirangi Rd., One Tree Hill. Phone 667-720 (h).
ATARI MICROCOMPUTER USERS GROUP: Brian or Dean Yakas. Phone 8363 060 (h). Meetings: Second Tuesday.
BBC USERS' GROUP: Dave Fielder. Phone 770-630 ext 516 (w).
BIG BOARD USER GROUP: Steve Van Veen, Flat 5, 111 Melrose Rd, Mt Roskill, Auckland 4. Phone (09) 659-991 (h).
BUSINESS USERS' GROUP: John Hawthorn, 11 Seaview Rd, Remuera. Phone 542-714 (h), 876-189 (w). Meetings monthly.
COMMODORE USERS' GROUP: Doug Miller, 18 Weldene Ave., Glenfield. Phone 444-9617 (h), 497-081 (w). Meetings: Third Wednesday.
CP/M USERS' GROUP: Kerry Koppert, 2/870 Dominion Rd., Balmoral. Phone 69-5355 (h). Meetings: Micro workshop.
DREAM 6800 USERS: Peter Whalan, 22 Kelston St, New Lynn, Auckland. Phone (09) 875110 (h).
KIM USERS: John Hirst, 1A Northboro Rd, Takapuna. Phone (09) 497-852 (h).
LNW USERS: Ray James. Phone (09) 30-839 (w), 595-587 (h).
SINCLAIR USERS' GROUP: Doug Farmer. Phone 667-589 (h). Meetings: Fourth Wednesday.
SORCERER USERS' GROUP (NZ): Selwyn Arrow. Phone 491-012 (h). Meetings: Micro workshop.
1802 USERS' GROUP: Brian Conquer. Phone 655-984 (h).
The above contacts can usually be found at NZ Microcomputer Club Meetings, or via P.O. Box 6210, Auckland.

Other Auckland-based groups:
ACES (Auckland Computer Education Society): Ray Clarke, 1 Dundas Pl., Henderson. Phone 836-9737 (h).
CMUG (Combined Microcomputer Users' Group): This is an association of Microcomputer Clubs, Groups, etc, formed to co-ordinate activities and to give a combined voice on topics concerning all micro users. Representation from all Clubs and Groups is welcomed to: CMUG C/- P.O. Box 6210, Auckland.
EPSON HX20 USERS' GROUP. Contact: C.W. Nighy, 14 Domett Avenue, Epsom, Auckland. (Answaphone, 774-268).
HP41C USERS' GROUP (Auckland): C/- Calculator Centre, P.O. Box 6044, Auckland. Grant Buchanan, 790-328 (w). Meets third Wednesday, 7pm, at Centre Computers, Great South Rd., Epsom.
NZ TRS-80 MICROCOMPUTER CLUB: Olaf Skarsholt, 203A Godley Rd., Titirangi. Phone 817-8698 (h). Meets first Tuesday, VHF Clubrooms, Hazel Ave., Mt Roskill, Auckland.
OSI USERS' GROUP (Ak): Vince Martin-Smith, 44 Murdoch Rd., Grey Lynn, Auckland. Meets third Tuesday, VHF Clubrooms, Hazel Ave., Mt Roskill.
SYMPOOL (NZ SYM USER GROUP): J. Robertson, P.O. Box 580, Manurewa. Phone 266-2168 (h).
A.Z..T.E.C.: Brian Mayo, Church Street, Katikati. Phone 490-326. Members use all micros and the club has just bought a Wizzard.
TAURANGA SINCLAIR COMPUTER CLUB: C. Ward, Secretary, P.O. Box 6037, Brookfield, Tauranga. Phone 82-962 or 89-234.
ATARI 400/800 USER CLUB: Dave Brown, P.O. Box 6053, Hamilton. Phone (071) 54-692 (h).
GISBORNE MICROPROCESSOR USERS' GROUP: Stuart Mullett-Merrick, P.O. Box 486, Gisborne. Phone 88-828.
ELECTRIC APPLE USERS' GROUP: Noel Bridgeman, P.O. Box 3105, Fitzroy, New Plymouth. Phone 80-216
TARANAKI MICRO COMPUTER SOCIETY: P.O. Box 7003, Bell Block, New Plymouth: Francis Slater. Phone 84-514.
HAWKE'S BAY MICROCOMPUTER USERS' GROUP: Bob Brady, Pirimai Pharmacy, Pirimai Plaza, Napier. Phone 439-016.
MOTOROLA USER GROUP: Harry Wiggins, (ZL2BFR), P.O. Box 1719, Palmerston North. Phone (063) 82-527 (h).
MICRO AND PEOPLE IN SOCIETY (MAPS): Levin, meets on second and fourth Thursday of each month. D. Cole, 28 Edinburgh Street, Levin. Phone 83-904, or W. Withell, P.O. Box 405, Levin.
ATARI USERS' GROUP, Wellington: Eddie Nickless. Phone 731-024 (w). P.O. Box 16011. Meetings: first Wednesday of month.
CENTRAL DISTRICTS COMPUTERS IN EDUCATION SOCIETY: Rory Butler, 4 John Street, Levin. (069) 84-466 or Margaret Morgan, 18 Standen Street, Karori, Wellington. (04) 767-167.
UPPER HUTT COMPUTER CLUB: Shane Doyle, 18 Holdworth Avenue, Upper Hutt. Phone 278-545. An all-machine club.
BBC USER GROUP: Users of other machines welcome too. Write P.O. Box 1561, Wellington, or Phone 861-213, Wellington.
OSBORNE USER GROUP: Dr Jim Baltaxe, C/- 75 Ghuznee Street, Wellington 1. Phone (04) 728-658.
NZ SUPER 80 USERS' GROUP: C/- Peanut Computers, 5 Dundee Pl., Chartwell, Wellington 4. Phone 791-172.
OHIO USERS' GROUP, Wellington. Secretary/Treasurer: R.N. Hislop, 658 Awatea Street, Porirua.
WELLINGTON MICROCOMPUTING SOCIETY INC.: P.O. Box 1581, Wellington, or Bill Parkin (h) 725-086. Meetings are held in Wang's Building, 203-209 Willis Street, on the 2nd Tuesday each month at 7.30pm.
NELSON MICROCOMPUTER CLUB: Dr Chris Feltham, Marsden Valley Rd, Nelson. Phone (054) 73-300 (h).
NELSON VIC USERS' GROUP: Peter Archer, P.O. Box 860, Nelson. Phone (054) 79-362 (h).
BLENHEIM COMPUTER CLUB: Club night second Wednesday of month. van Meynell, Secretary, P.O. Box 668. Phone (h) 85-207 or (w) 87-834.
CHRISTCHURCH ATARI USERS GROUP: Contact Edwin Brandt. Phone 228-222 (h), 793-428 (w).
CHRISTCHURCH '80 USERS' GROUP: David Smith, P.O. Box 4118, Christchurch. Phone 83-111 (h).
CHRISTCHURCH PEGASUS USERS' GROUP: Don Smith, 53 Farquhars Rd, Redwood, Christchurch. Phone (03) 526-994 (h), 64-544 (w), ZL3AFP.
CHRISTCHURCH APPLE USERS' GROUP: Paul Neiderer, C/- P.O. Box 1472, Christchurch. Phone 796-100 (w).
OSI USERS' GROUP (CH): Barry Long, 377 Barrington St., Spreydon, Christchurch. Phone 384-560 (h).
CHRIST CHURCH SINCLAIR USERS' GROUP: Mr J. Mitchell. Phone 385-141, P.O. Box 33-098.
CHRISTCHURCH COMMODORE USERS' GROUP: John Kramer, 885-533 and John Sparrow, Phone 896-095.
ASHBURTON COMPUTER SOCIETY: Mr J. Clark, 52 Brucefield Avenue.
SOUTH CANTERBURY COMPUTERS' GROUP: Caters for all machines for ZX81 to IBM34. Geoff McCaughan. Phone Timaru 84-200 or P.O. Box 73.
LEADING EDGE HOME COMPUTER CLUB: Elaine Orr, Leading Edge Computers, P.O. Box 2260, Dunedin. Phone 55-268 (w).
DUNEDIN VIC USERS' GROUP: Terry Shand, 24 Bremner Road, Fairfield. Phone (024) 881-432. Meetings last Thursday of month.
DUNEDIN SORD USERS' GROUP: Terry Shand. Phone (024) 771-295 (w), 881-432 (h).
NOTE: Clubs would appreciate a stamped, self-addressed envelope with any written inquiry to them.

NOTE: If your club or group is not listed, drop a line with the details to: Club Contacts, BITS & BYTES, Box 827, Christchurch. The deadline for additions and alterations is the second weekend of the month before the next issue.

NEW PRODUCTS

A portable computer to rival the Osborne 1 is now on sale in New Zealand.

Like the Osborne the Kaypro 11 folds into a carrying case, although at 26 pounds it is two pounds or approximately one kilogram heavier than the Osborne.

The Kaypro 11 has a Z-80 microprocessor, 64K RAM, a 76 key detachable keyboard (with numeric keypad), an RS-232C interface port and centronics parallel printer port and twin 5¼ inch floppy disk drives with 191K of user memory on each.

Probably the most significant difference from the Osborne is the screen. The Kaypro 11 has a 22.9 cm (9 inch) screen (measured on the diagonal) displaying 80 characters wide by 24 lines compared to the Osborne's 12.7 cm (5 inch) screen.

Also like the Osborne several software packages are included as standard in the Kaypro 11's $4690 price tag. These are CP/M disk operating system, S-BASIC (the "S" stands for structured), a word processing program called Select (and a disk called Teach which explains word processing to the first time user) and Profitplan, a financial/calculation spreadsheet program again said to be oriented to the first time user.

New Zealand agents for the Kaypro are President Computers of Auckland. We will have more on the Kayporo 11 in a later issue.

GLOSSARY

BASIC: Beginners' All-purpose Symbolic Instruction Code. The most widely used, and easiest to learn, high level programming language (a language with English-like instructions) for microcomputers.

Binary: The system of counting in 1's and 0's used by all digital computers. The 1's and 0's are represented in the computer by electrical pulses, either on or off.

Bit: Binary digit. Each bit represents a character in a binary number, that is either a 1 or 0. The number 2 equals 10 in binary and is two bits.

Boot: To load the operating system into the computer from a disk or tape. Usually one of the first steps in preparing the computer for use.

Bug: An error in a program.

Byte: Eight bits. A letter or number is usually represented in a computer by a series of eight bits called a byte and the computer handles these as one unit or "word".

Character: Letters, numbers, symbols and punctuation marks each of which has a specific meaning in programming languages.

Chip: An integrated circuit etched on a tiny piece of silicon. A number of integrated circuits are used in computers.

CP/M: A disk operating system available for microcomputers using a particular microprocessor (that is the 8080 and Z80 based microcomputers such as the TRS 80 and System 80). See also Disk Operating Systems.

Cursor: A mark on a video that indicates where the next character will be shown, or where a change can next be made.

Data: Any information used by the computer either I/O or internal information. All internal information is represented in binary.

Disk: A flat, circular magnetic surface on which the computer can store and retrieve data and programs. A flexible or floppy disk is a single 8 inch or 5¼ inch disk of flexible plastic enclosed in an envelope. A hard disk is an assembly of several discs of hard plastic material, mounted one above another on the same spindle. The hard disk holds up to hundreds of millions of bytes – while floppy disks typically hold between 140,000 and three million bytes.

Disk drive: The mechanical device which rotates the disk and positions the read/write head so information can be retrieved or sent to the disk by the computer.

Disk operating system: A set of programs that operate and control one or more disk drives. See CP/M for one example. Other examples are TRSDOS (on TRS 80) and DOS 3.3 (for Apples).

Firmware: Programs fixed in a computer's ROM (Read Only Memory); as compared to software, programs held outside the computer.

Hardware: The computer itself and peripheral machines for storing, reading in and printing out information.

Input: Any kind of information that one enters into a computer.

Interface: Any hardware/software system that links a microcomputer and any other device.

K: The number 1024. Commonly refers to 1024 bytes. Main exception is capacity of individual chips, where K means 1024 bits.

KILOBYTE (or K): Represents 1024 bytes. For example 5K is 5120 bytes (5 x 1024).

Machine language: The binary code language that a computer can directly "understand".

Megabyte (or Mb): Represents a million bytes.

Memory: The part of the microcomputer that stores information and instructions. Each piece of information or instruction has a unique location assigned to it within a memory. There is internal memory inside the microcomputer itself, and external memory stored on a peripheral device such as disks or tape.

Microcomputer: A small computer based on a microprocessor.

Microprocessor: The central processing unit or "intelligent" part of a microcomputer. It is contained on a single chip of silicon and controls all the functions and calculations.

Network: An interconnected group of computers or terminals linked together for specific communications.

Pascal: A high-level language that may eventually rival BASIC in popularity.

PEEK: A command that examines a specific memory location and gives the operator the value there.

Peripherals: All external input or output devices: printer, terminal, drives etc.

Pixel: Picture element. The point on a screen in graphics.

POKE: A command that inserts a value into a specific memory location.

Program: A set or collection of instructions written in a particular programming language that causes a computer to carry out or execute a given operation.

RAM: Random access memory. Any memory into which you "read" or call up data, or "write" or enter information and instructions.

REM statement: A remark statement in BASIC. It serves as a memo to programmers, and plays no part in the running program.

ROM: Read only memory. Any memory in which information or instructions have been permanently fixed.

Software: Any programs used to operate a computer.

System: A collection of hardware and software where the whole is greater than the sum of the parts.

Tape: Cassette tape used for the storage of information and instructions (not music).

Word: A group of bits that are processed together by the computer. Most microcomputers use eight or 16 bit words.

Verbatim Datalife® flexible disks now come in a bold, new storage box. But more important, they now come to you with a five year warranty*.

We can give you a warranty this long because we're confident the way we make Datalife disks will make them perform better, last even longer.

All of our Datalife disks feature seven data-shielding advances for greater disk durability, longer data life. To protect your data from head-to-disk abrasion. To shield your data against loss due to environmental conditions. To ensure a longer lifetime of trouble-free data recording, storage and retrieval.

Every Datalife disk is extensively tested under the most extreme conditions. Critically-certified to be 100% error-free. Assuring you an added margin of performance, no matter what the operating conditions.

And we back it up with a five years warranty. Five times longer than the industry standard. Because Verbatim is the standard of excellence.

Verbatim®

Here's the most exciting part of Verbatim's new packaging.

5 year warranty

For flexible disks you can depend on — a lot longer, call (09) 444-6085

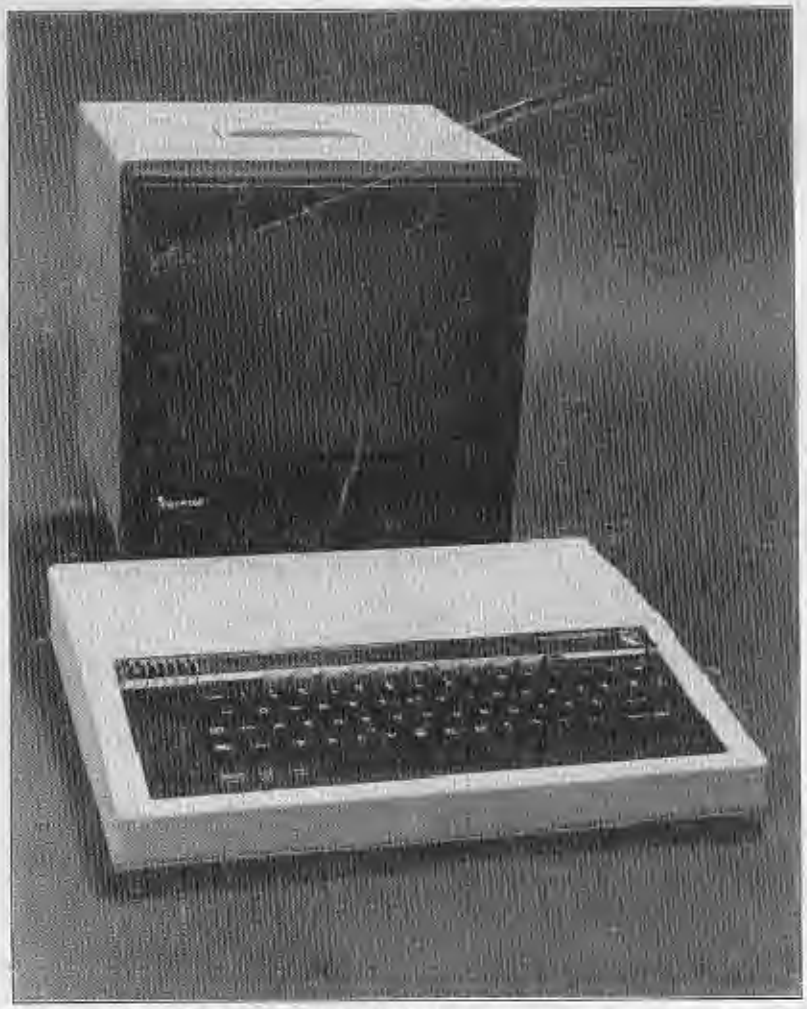
COMPUDATA MEDIA SYSTEMS LTD
Box 3273, 48 Ellice Road, Glenfield, Auckland

Here is the news . . . .

The B.B.C. has arrived

"Firstly the machine has been laid out with a clear growth path in mind: you can see how the design has room for expansion.
Secondly the machine as it stands is highly flexible. It possesses the 'openness' of the Apple plug-it-in peripheral card philosophy, the screen software flexibility of the Atari 400 and 800 machines and the teletext compatibility of Poly. Finally it seems to have taken all the little edges that most current machines possess, rounded them off and then French polished them."

Dr. P. FORER in Bits and Bytes
November, 1982



B.B.C. Micro

The face of things to come!

FACTS THAT MAKE THE BBC BETTER

- Price. . . is competitive. . . **$1995**
- Good Colour. . . essential for entertainment and in presenting information, you can highlight different areas at the same time.
- The resolution of the screen allows 640 dots x 256 dots which makes the BBC the best system available at this price. When making Graphs and curves, the better the resolution the clearer the curve.
- Sound. . . four sound channels, three note generators, one noise generator. Full sound envelope control on sound channels. Is adaptable to have a voice synthesizer fitted. The production of the sound does not stop or delay the running of the central process.
- Basic. . . allows the use of PROCEDURES and REPEAT UNTIL loops, which give it the advantages of a structured language.
- The BBC model B is the computer of the future. . . It is able to be hooked up to a second processor when necessary, the Z80 and 6502 already released. The BBC will never go out of date! A.16 Bit Processor to be released soon.

- ☆ **EDUCATION**
- ☆ **TRAINING COURSES AVAILABLE**
- ☆ **ON SITE DEMONSTRATIONS**
- ☆ **HOME DEMONSTRATIONS**
- ☆ **FULL TECHNICAL ADVISORY SERVICE**
- ☆ **GAMES/ENTERTAINMENT**

Distributed by:

access **Access Data Ltd Auckland Phone 686 578**

Branches in CHRISTCHURCH and WELLINGTON — and authorised dealers.